Knowledge acquisition from user reviews for Interactive Question Answering

Natalia Konstantinova

A thesis submitted in partial fulfilment of the requirements of the University of Wolverhampton for the degree of Doctor of Philosophy

2013

This work or any part thereof has not previously been presented in any form to the University or to any other body whether for the purposes of as- sessment, publication or for any other purpose (unless otherwise indicated). Save for any express acknowledgements, references and/or bibliographies cited in the work, I confirm that the intellectual content of the work is the result of my own efforts and of no other person.

Signature: . . . . . .

Date: . . . . . .

Abstract

Nowadays, the effective management of information is extremely important for all spheres of our lives and applications such as search engines and question answering systems help users to find the information that they need. However, even when assisted by these various applications, people sometimes struggle to find what they want. For example, when choosing a product customers can be confused by the need to consider many features before they can reach a decision. Interactive question answering (IQA) systems can help customers in this process, by answering questions about products and initiating a dialogue with the customers when their needs are not clearly defined.

The focus of this thesis is how to design an interactive question answering system that will assist users in choosing a product they are looking for, in an optimal way, when a large number of similar products are available. Such an IQA system will be based on selecting a set of characteristics (also referred to as product features in this thesis), that describe the relevant product, and narrowing the search space. We believe that the order in which these characteristics are presented in terms of these IQA sessions is of high importance. Therefore, they need to be ranked in order to have a dialogue which selects the product in an efficient manner.

The research question investigated in this thesis is whether product

characteristics mentioned in user reviews are important for a person who is likely to purchase a product and can therefore be used when designing an IQA system.

We focus our attention on products such as mobile phones; however, the proposed techniques can be adapted for other types of products if the data is available. Methods from natural language processing (NLP) fields such as coreference resolution, relation extraction and opinion mining are combined to produce various rankings of phone features.

The research presented in this thesis employs two corpora which contain texts related to mobile phones specifically collected for this thesis: a corpus of Wikipedia articles about mobile phones and a corpus of mobile phone reviews published on the Epinions.com website. Parts of these corpora were manually annotated with coreference relations, mobile phone features and relations between mentions of the phone and its features.

The annotation is used to develop a coreference resolution module as well as a machine learning-based relation extractor. Rule-based methods for identification of coreference chains describing the phone are designed and thoroughly evaluated against the annotated gold standard. Machine learning is used to find links between mentions of the phone (identified by coreference resolution) and phone features. It determines whether some phone feature belong to the phone mentioned in the same sentence or not.

In order to find the best rankings, this thesis investigates several settings. One of the hypotheses tested here is that the relatively low results of the proposed baseline are caused by noise introduced by sentences which are not directly related

to the phone and phone feature. To test this hypothesis, only sentences which contained mentions of the mobile phone and a phone feature linked to it were processed to produce rankings of the phones features. Selection of the relevant sentences is based on the results of coreference resolution and relation extraction.

Another hypothesis is that opinionated sentences are a good source for ranking the phone features. In order to investigate this, a sentiment classification system is also employed to distinguish between features mentioned in positive and negative contexts.

The detailed evaluation and error analysis of the methods proposed form an important part of this research and ensure that the results provided in this thesis are reliable.

Acknowledgements

Completing my doctoral studies was more difficult than I anticipated, however, I was able to face this challenge successfully because I was surrounded by great people who supported me for all these years.

The biggest "thank you" goes to my family, my parents and my brother. I cannot possibly express with words how much their support and love means for me and how grateful I am for their efforts to give me the best in life. Thank you very much for always being there for me!

I am thankful to Prof. Ruslan Mitkov, my director of studies, for guiding me in the scientific world, providing me with support for all the years of my PhD and for giving me an opportunity to learn a lot about NLP by being part of the Journal of Natural Language Engineering and RANLP conferences. A big "thank you" to my supervisor and friend, Dr. Constantin Orăsan, who made me believe that I can achieve more than I initially thought I was capable of. I am grateful for his reliability, willingness to help in all situations and extensive feedback for my research. I wish also to thank Dr. Marco de Boni for making this PhD possible and Unilever for providing partial financial support for my scholarship.

I am grateful to my examiners Prof. Ricardo Mairal Usón and Prof. Michael Thelwall for taking the time to read my thesis and for all their insightful comments. Special "thank you" to Ricardo for managing to come to my viva in spite of the

airport strike. I wish also to thank my former supervisor, Olga Mitrofanova, for encouraging me to come to RIILP. I would like to thank the Bulgarian team (Galia Angelova, Ivelina Nikolova and Irina Temnikova) who were involved in the organisation of RANLP that became an integral part of my PhD studies.

I would like to thank all current and former members of my group who became my friends over the years. I am very grateful to each of you for the friendly environment and being able to cheer up and joke in all situations. An additional "thank you" to those who had the patience to proof-read my work over the years: Alison Carminke, Daniel Clayton, Emma Franklin and Erin Stokes. Special thanks to Emma for going through my thesis so quickly. I am also very grateful for the administrative support I received from Alison Carminke, Emma Franklin, Stephanie Kyle and Erin Stokes.

Finally, a big "thank you" to all my friends who were able to understand the big gaps between the emails we exchanged and who encouraged me all the way throughout my PhD. Special "thank you" to Daniel for being very patient and understanding in the most stressful period of my studies. Also great thanks for the encouragement and belief in me and, of course, for those many bars of dark chocolate that supported me all the way through long working hours. I would also like to thank my great friends that I met during my stay in UK and who taught me some valuable lessons: Simona Damyanova, Paul Joy, Jean Kaplan, Natalia Ponomareva and Sanja Štajner.

I was very lucky to be surrounded by such amazing people, hope I did not forget to mention anyone. Thank you for your belief in me and my ability to succeed!

Contents

List of Tables

List of Abbreviations

ACE – Automatic Content Extraction

AI – Artificial Intelligence

AIML – Artificial Intelligence Markup Language

CBPS – Constraint-Based Problem-Solver

CLEF – Cross-Language Evaluation Forum

ComplInfo – Complementary Info

CoreInfo – Core Information

CRF – Conditional Random Fields

DAARC – Discourse Anaphora and Anaphor Resolution Colloquium

DIPRE – Dual Iterative Pattern Relation Expansion

DM – Dialogue Manager

DS – Dialogue System

EAT – Expected Answer Type

FQ – Follow-up Question

IE – Information Extraction

IQA – Interactive Question Answering

IR – Information Retrieval

IWP – Intelligence in Wikipedia

ML – Machine Learning

MT – Machine Translation

MUC – Message Understanding Conference

NE – Named Entity

NER – Named Entity Recognition

NL – Natural Language

NLG – Natural Language Generation

NLP – Natural Language Processing

NLTK – Natural Language Toolkit

NLU – Natural Language Understanding

NP – Noun Phrase

Tau – Kendall's Tau

TE – Terminology Extraction

TF – Term Frequency

TREC – Text REtrieval Conference

QA – Question Answering

POS – Part of Speech

RDF – Resource Description Framework

Rho – Spearman's Rank Correlation Coefficient or Spearman's Rho

VP – Verb Phrase

URL – Uniform Resource Locator

XML – eXtended Markup Language

Chapter 1

Introduction

1.1 Overview

Nowadays, the effective management of information is extremely important for all spheres of our lives. There are different applications, such as information retrieval and question answering systems, which may help users to find information. However, even when assisted by these various applications, people sometimes struggle to find what they want. Users may have in mind what they need and what they are looking for, but fail to formulate their requests properly, and, therefore, are unable to search for information efficiently. In everyday communication, speakers are involved in dialogues, where they have the chance to elaborate and clarify their utterances. However, when dealing with an automatic system, users lack this kind of natural interaction. Therefore, they cannot explain their needs, because providing the system with enough information from the first interaction can be quite difficult.

Naturally, the success of the search also greatly depends on the type of questions involved. For example, answering a question such as *"Where is Paris situated?"* can usually be processed with relative ease. The only problem lies with the ambiguity of the word *"Paris"*, which can refer to several different entities in

the real world. However, if the users need to find a phone with a GPS and a 1.5 megapixel camera, the query becomes more complex, and additional questions may be required to clarify their request. Interactive question answering (IQA) systems can help customers in this process by answering questions about products and initiating a dialogue with customers when their needs are not clearly defined. When choosing a product, customers can be confused because they need to consider many features before they can reach a decision. The possibility to ask additional questions allows users to refine their queries step by step if needed. Therefore the use of an IQA system can facilitate the search for information and make it more effective.

The focus of this thesis is how to design an interactive question answering system which will assist users in choosing from many similar products the one they are looking for, in an optimal way. We are seeking ways to facilitate interaction involving complex questions and make the search for information easier and quicker.

Such an IQA system will be based on selecting a set of characteristics (also referred to as product features) which describe the relevant product and narrow the search space. We believe that the order in which these characteristics are presented in terms of these IQA sessions is of high importance. Therefore, they need to be ranked in order to have a dialogue which selects the product in an efficient manner.

1.2 Domain

Firstly, we should point out that we aim to design a system intended to work within a well defined domain. It is motivated by our belief that it is not feasible to build a highly accurate open domain system, and therefore domain restricted systems are preferable. IQA systems aim to assist information seeking, therefore restricting the domain can make them more precise. However, as a result of our research, we will try to design a general methodology which can be later adapted and used for other domains.

For our research, we decided to focus on the domain of products. In order to agree on how to define the domain more precisely, we need to agree what constitutes a "product". We believe that all objects of the real world which can be referred to as "products" usually have a set of common characteristics. All products have a name and a set of features which differentiate them from other models and varieties of similar products, and yet within a series there are several examples with similar features. These characteristics define, for example, that all products of a given series have common features and at the same time have something that distinguishes one model from another. All these characteristics make it possible to adopt the constraint management approach for IQA (described in more detail in Chapter 2). This approach is based on the fact that a question is treated by the system as a list of constraints and the answer is a range of objects satisfying these constraints.

In the modern world of rapidly developing technologies and online markets,

an IQA system which can assist a customer in choosing a product seems very promising and in high demand. Frequently people want to buy a product, but fail to do so, because they do not manage to choose the appropriate one. During their search, customers can also waste a lot of their time reading specifications and reviews. As a result, too many options leave the user frustrated by the overwhelming amount of information.

Companies are interested in developing newer and better products, but they leave their consumers with an enormous amount of choice, which, in the long run, prevents them from buying products. Psychological studies Schwartz (2004) reveal that the axiomatic belief that "choice is good, and that more choice is better" is not true, and people feel happier if they have some restriction on the choices available. Therefore, we believe that the domain of products will greatly benefit from the use of IQA systems. The constraint-based approach will help to present information in an efficient way and assist the process of decision making.

As a starting point for our research, we have chosen a subdomain of the product domain – the domain of mobile phones. However, as previously mentioned, we will attempt to develop methods and techniques that can be generalised and used for other subdomains of products.

Our choice of the domain of mobile phones was motivated by several factors:

- this domain is quite restricted and specific (but not too specific);
- the domain is a subdomain of "products";
- there is a need for IQA systems in this domain.

Each of these factors are briefly explained below.

The domain of mobile phones is restricted and, once we decide what devices can be referred to as "phones", its boundaries can be clearly defined. However, it is not very narrow and, therefore, can feature different kinds of information.

The domain of phones bears similarities to other domains such as the ones describing cars, computers, etc. These objects are united by one common characteristic – they are all products which have several features to differentiate them. Therefore, as mentioned earlier, a special structure of the product domain can be used when building a resource for an IQA.

Last but not least, there is a real need for IQA systems in the domain of mobile phones. Every day millions of people choose a new phone, and as technologies advance, this process is becoming more and more difficult. The amount of features characterising a mobile phone can be enormous. A growing number of people have started using the Internet regularly when choosing products; however, the amount of information available makes it difficult to search. Thus, the need for assistance is increasing. Businesses, e.g. online shops, are also extremely interested in providing effective technologies to outperform their competitors. Therefore, IQA systems in the domain of mobile phones is in high demand, and both businesses and customers can benefit using it.

1.3 Example of IQA dialogue

An example of a typical dialogue which could occur between a shop assistant and a customer choosing a phone is presented in this section. The user wants to buy

a phone which has a camera and a GPS, but he/she has not decided yet whether to have a touch screen or which brand to choose:

(1) ASSISTANT: Hi!
(2) CUSTOMER: Hello, can you help me to find a smart phone with a camera?
(3) ASSISTANT: Sure! There are [BIG NUMBER] of smart phones featuring a camera, would you like to look for some particular [BRAND]?
(4) CUSTOMER: No, which one has GPS?
(5) ASSISTANT: [STILL BIG NUMBER] phones have GPS, are you interested in ones having TOUCH SCREEN?[Suggesting new constraints]
(6) CUSTOMER: Yes, it would be great.
(7) ASSISTANT: We have [NUMBER] of Nokia phones, [NUMBER] HTC phones, [NUMBER] Samsung phones, [NUMBER] of other brands matching your requirements.
(8) CUSTOMER: Ok, what about HTC phones?
(9) ASSISTANT: [NUMBER] of them have [feature_1], [NUMBER] of them have [feature_2].
(10) CUSTOMER: What are the ones with [feature_1]?
(11) ASSISTANT: You can choose from: [RESULTS].

This dialogue looks natural and can be accommodated by an IQA system. However, if we attempt to have a similar dialogue using a classical question answering (QA) system, we will face several problems. Assuming that the user knows from the beginning all the features the phone he/she is looking for needs to have, his/her request to the QA system would look like: "What HTC smart phones have camera, feature_1, GPS, touch screen...?". In addition to the QA system's difficulty in processing the query, this kind of question also looks very unnatural. Moreover, given the customer is not sure about his/her needs, it is

difficult to formulate a question from the beginning and take into account all the wishes of the user. Users can also change their mind when presented with the intermediate results. For this reason, the interaction is needed in order to refine the query. Therefore, tackling this situation when using a classical QA system does not seem feasible.

An IQA system should be able to handle the dialogue presented above. For example, the IQA system can help by suggesting new constraints (such as in steps (3) and (5)) at the point when users have too many results returned to their query. Users can agree with the suggested constraints as in step (6) or reject them and think of their own as in step (4). Interactivity can also benefit from presenting results already classified due to some constraints, e.g. in line (7) the system decides to sort all the results according to their brands. In this way, it suggests another constraint that can be chosen by the user. This example of a hypothetical IQA session shows that interaction can make the search easier and more intuitive for the user.

1.4 Goals and contributions

The main **aim** of this thesis is to identify ways to design an interactive question answering system which will assist users in choosing a product they are looking for in an optimal way when a large number of similar products are available. We are seeking methods which can facilitate interaction involving complex questions and make the search for information easier and quicker.

This thesis is based on several hypotheses that are tested and evaluated in our

research:

- **Hypothesis 1**: It is possible to use a data-driven approach for designing an IQA system.
- **Hypothesis 2**: Product characteristics mentioned in user reviews are important for a person who is likely to purchase a product and can therefore be used when designing an IQA system. A ranking of product features can be used to provide an IQA system with information on which product characteristics should be given priority and presented first.
- **Hypothesis 3**: Only sentences which are directly related to the phone and phone feature should be used for the ranking of the features in order to reduce noise introduced by other sentences.
- **Hypothesis 4**: Opinionated sentences are a good source when it comes to ranking the phone features.

In order to achieve the main aim of this thesis and test the hypotheses, several **goals** need to be met:

- **Goal 1** is to investigate the field of IQA and review the current approaches to the design of IQA systems. It is necessary to identify which methods are the most relevant to build an IQA system.
- **Goal 2** is to collect and annotate resources to be used for this research. The research presented in this thesis employs two corpora which contain texts related to mobile phones specifically collected for this thesis: a corpus of Wikipedia articles about mobile phones and a corpus of mobile phone

reviews published on the Epinions.com website. Parts of these corpora were manually annotated with coreference relations, mobile phone features and relations between mentions of the phone and its features.

- **Goal 3** is to develop a coreference resolution module which identifies all mentions of the product in the text. **Goal 4** is to develop a machine learning-based relation extractor, which is used to find links between mentions of the phone (identified by coreference resolution) and phone features. These goals need to be achieved in order to test one of the hypotheses of this research, specifically, that sentences which are not directly related to the phone and its features can introduce noise in the processing.
- **Goal 5** is to investigate whether product characteristics mentioned in user reviews are important for a person who is likely to purchase a product and can therefore be used when designing an IQA system. We explore different ways of ranking product features in order to be able to provide an IQA system with information on which product characteristics should be given priority and presented first.

1.5 Original contributions

By achieving goals mentioned in the previous section, this research makes several contributions to the fields of information extraction and interactive question answering. To sum up, the original contributions of this thesis are listed below.

- **The first original contribution** is the development of a new approach to enhance the work of IQA. The suggested strategy of ranking features in terms of IQA systems developed specifically for product search is a novel one, and, to the best of our knowledge, was first suggested in this thesis. In order to find the best rankings, this thesis investigates several settings. To evaluate these settings, a gold standard is collected using input from human respondents. All the rankings that are produced using different methods are compared against this gold standard. This approach is also tested using extrinsic evaluation.
- **The second original contribution** is the development of a rule-based coreference resolution, which links all mentions of the phone in the text. Coreference resolution was developed for two types of texts: Wikipedia texts and review texts, describing mobile phones. The designed methods for identification of coreference chains are thoroughly evaluated against the annotated gold standard. The suggested approach can be also used for other domains of products, as long as some world knowledge relevant to the new domain is added to the algorithm.
- **The third original contribution** is the development and evaluation of a machine-learning approach to discover links between the mentions of the phones and their features. This method determines whether a phone feature belongs to the phone mentioned in the same sentence or not. Different machine learning features inspired by relation extraction literature are explored to find the combination that yields the best results.

- **The fourth original contribution** represents the resources compiled for the experiments carried out in this thesis:
  - a corpus of Wikipedia articles describing mobile phones (texts and infoboxes);
  - a review corpus describing mobile phones;
  - a corpus of Wikipedia articles annotated with markables and coreference chains for the main topic;
  - a corpus of review articles annotated with coreference for the main topic and features of the phone;
  - a corpus of review texts annotated with links between mentions of the phone and features;
  - a database of phones and their features compiled using Wikipedia infoboxes.

  The compilation process of the first two corpora is detailed in Chapter 3. The coreference corpora are described in Section 4.4.2. The creation of the corpus annotated for links is discussed in Section 5.3. The last resource is an additional output of the work carried out in Section 6.5.

Another contribution of this thesis is a literature review in the field of IQA. This field is a relatively new one, and lacks general articles/book chapters describing the state of the art and recent advances in the area. The literature review discusses the state of the art in IQA and presents the background information from QA and dialogue systems needed to understand basic concepts of IQA.

1.6 Structure of the thesis

This thesis comprises seven chapters in total. Chapter 2 provides background for research in the interactive question answering (IQA). Whereas, chapters 3 to 6 constitute the original contribution of this thesis.

Chapter 2 provides the background information needed to properly understand the goals of this research. It presents basic notions and describes the state of the art in the field of interactive question answering. It also emphasises the potential of IQA for addressing the problem of effective search for information while highlighting the lack of research in this field.

Chapter 3 describes the data characteristics and presents the steps undertaken to prepare it for the experiments described in the chapters to follow. Specifically, two types of corpora are discussed: semi-structured texts and unstructured texts. The steps taken to collect these corpora are also detailed in this chapter. Both corpora are used for the development of coreference resolution methods described in Chapter 4. The second corpus is also used for the identification of the links between mentions of the phone in the text and features. This task is described in Chapter 5. Chapter 3 also discusses the extraction of features describing mobile phones using two types of approaches: corpus-based approaches and methods which exploit the structure existing in resources.

Chapter 4 addresses the problem of coreference resolution for two types of corpora. First, it provides background in the field of information extraction and coreference resolution. Then, it discusses the motivation to develop our own

coreference resolution system rather than use already existing systems. This chapter covers the annotation guidelines and the process of corpus annotation, as well as the development of the coreference resolution algorithm. The details about the system's evaluation are followed by the thorough error analysis of the algorithm suggested for coreference resolution.

Chapter 5 tackles the problem of identification of links between the mentions of the phone and its features. It starts with a brief overview of the state of the art of relation extraction and the description of existing ways of approaching this task. This is followed by the discussion of the corpus annotation to prepare a gold standard. This chapter also provides a detailed description of machine learning (ML) features used for our algorithm and the way they were extracted. The 5-fold cross-validation of different combinations of features is presented as well. We also examine the errors in the classification output and try to identify the reason for the algorithm failing to classify sentences correctly.

Chapter 6 addresses the problem of feature ranking for interactive question answering systems which help customers to choose the right product for them. First, the relevant work in the field is discussed. It is followed by the description of the experiment including its justification and ranking methods developed. Two types of evaluation are carried out to test the performance of our method: intrinsic and extrinsic evaluation. Error analysis and discussion of the results is provided for both types of evaluation.

Chapter 7 revisits goals of this thesis and summarises the original contributions of this research. It also provides a review of the thesis and presents

the main conclusions which can be drawn from the investigation carried out in the previous chapters. This is followed by the indications of the directions for future research.

Chapter 2

Interactive Question Answering

2.1 Overview

This chapter provides background information that is necessary to properly understand the goals of this research. It presents basic notions and describes the state of the art of the field.

Interactive question answering (IQA) is a research field that emerged at the intersection of question answering and dialogue systems. IQA inherits from Question Answering (QA) the features that allow users to ask questions in natural language and, where possible, locate the actual answer to the question. The interactive aspect of the field comes from the fact that a dialogue can be initiated with a user in cases where there are too many or too few answers, or there is some ambiguity in the request. The IQA systems also allow users to ask additional questions if the obtained result is not really what they are looking for or in cases where they need more information. For this reason, Webb and Webber (2009) define IQA as a *"process where the user is a continual part of the information loop"*.

Despite the wide variations in the ways different IQA systems are implemented, they generally rely on a scaled-down version of a dialogue system or at least on

some components of these systems. For this reason, the basic concepts behind dialogue systems (Section 2.2) and their structure (Section 2.3) are first described. Given the importance of question answering systems in the context of IQA, a brief introduction to question answering is also included (Section 2.4). The longest part of this section is dedicated to what IQA is (Section 2.5), including the most important approaches used by the IQA systems (Section 2.6), followed by the challenges that need to be faced when such systems are evaluated (Section 2.7).

2.2 Dialogue systems

The term *dialogue system* is widely used nowadays to refer to automatic systems involving coherent dialogue with a human interlocutor. Editors of the Journal of Dialogue Systems define a dialogue system as:

> a computational device or agent that (a) engages in interaction with other human and/or computer participant(s); (b) uses human language in some form such as speech, text, or sign; and (c) typically engages in such interaction across multiple turns or sentences (http://www.jods.org/).

This definition highlights several important aspects of such systems. A dialogue system always has a user, who interacts with the system towards a specific goal such as completing some tasks. The interaction involves a conversation in human language between two or more participants and can take several turns. The fact that human language is used in the interaction differentiates this field from others

such as database access using computer languages (such as SQL) or interaction between software agents that communicate using XML or other standard computer formats.

Early versions of dialogue systems were referred to as *chatterbots* or *chatbots* (Mauldin, 1994) indicating their rather simple goal. The initial role of these systems was to fool users into thinking they were communicating with humans in an attempt to replicate the Turing test (Turing, 1950). However, as the field progressed, the interest in dialogue systems shifted from pure academic research to commercial applications of the technology. Some dialogue systems, referred to as *conversational agents* by some researchers (Lester et al., 2004; Jurafsky and Martin, 2009), are used by companies in fields such as customer service, helpdesks, website navigation, guided selling and technical support. This is thanks to the fact that they offer a natural way of interacting with a computer, meaning that usually dialogue system users do not need any special training as the systems are easy to use and intuitive.

2.3 The architecture of dialogue systems

The first dialogue systems, the chatterbots, had a very simple architecture and relied only on pattern matching and the presence of particular keywords in the human utterances to produce an output. Whilst the first few turns of a conversation with a chatterbox may seem fine, the interaction often quickly degrades into nonsense. For this reason, current dialogue systems rely on more complex processing and have several modules. The structure of a dialogue system

varies a lot from one system to another, but Jurafsky and Martin (2009) consider that they usually consist of 5 main components: *speech recognition, natural language understanding (NLU), dialogue management, natural language generation (NLG) and speech synthesis.* Some of them are optional and can be absent in some systems. For example, the speech recognition and speech synthesis modules can be omitted due to the additional challenges they pose, despite offering a more natural way of interacting with the system. The aim of the NLU module is to produce a semantic representation appropriate for a dialogue task, whereas the NLG component is responsible for automatically creating natural language that is shown to the user on the basis of representation received. In this research, we address only the dialogue management component, so it is described in more detail later. However more information about NLG and NLU components can be found in (Konstantinova and Orăsan, 2013).

2.3.1 Dialogue management component

The Dialogue Manager (DM) is one of the most important parts of a dialogue system as it *coordinates the activity of several subcomponents of the dialogue system and its main goal is to maintain a representation of the current state of the ongoing dialogue* (Bui, 2006). Traum and Larsson (2003) identify the main tasks of the DM as:

- updating the dialogue context on the basis of interpreted communication;
- providing context-dependent expectations for interpretation of observed signals as communicative behaviour;

- interfacing with task/domain processing (e.g. database, planner, execution module, other back-end systems), to coordinate dialogue and non-dialogue behaviour and reasoning and
- deciding what content to express next and when to express it.

They also point out that in many systems some of these functions can also be delegated to other components of the system.

The dialogue manager has to interpret the speech acts, carry out problem-solving actions, formulate responses and in general maintain the system's idea of the state of the discourse (Dale et al., 2000). Therefore, the dialogue manager controls the whole architecture and structure of the dialogue and also serves as an interlink between the NLU and NLG components, as it takes information from the former, transforms it and passes to the latter.

Many different ways to classify dialogue managers can be found in the literature (Varges et al., 2007; Bui, 2006; Catizone et al., 2002; Xu et al., 2002). However, discussion of different types of dialogue managers is beyond the scope of this thesis.

2.4 Question answering

Question answering is defined as

> *an interactive process that encompasses understanding a user's information need, typically expressed in a natural language query; retrieving relevant documents, data, or knowledge from selected sources; extracting, qualifying and prioritizing available answers from these*

sources; and presenting and explaining responses in an effective manner

(Maybury, 2004)

Despite being defined as an interactive process, most of the existing QA systems limit this interaction to giving the user the possibility of asking one question and providing one or several answers without the possibility of any further communication. The existing research has focused mainly on factoid question answering where the user asks one question and the answer is a simple fact which is usually a named entity like a person, organisation or location. Systems that answer other types of questions, such as definition questions (Blair-Goldensohn et al., 2004; Cao and Huang, 2008), why questions (Verberne et al., 2010, 2011) and complex questions (Bilotti and Nyberg, 2006; Xiaoming and Li, 2010), were also investigated. This section presents only a brief introduction to QA in order to facilitate the understanding of the rest of the chapter. A comprehensive description of the QA field can be found in (Harabagiu and Moldovan, 2003; Webber and Webb, 2010).

A standard QA system consists of a pipeline of three modules: question processor, document processor and answer extractor (Harabagiu and Moldovan, 2003). The role of the question processor is to use various NLP techniques to interpret the question. This interpretation normally involves determining the semantic category of the answer (e.g. person, organis ation, location, etc.) and extracting a set of keywords from the question in order to produce a list of query terms for the document processing module. Advanced question processors try to build a more elaborate interpretation of the question, by adding semantic

information to these terms and identifying syntactic relations between them. To a certain extent, the question processor performs a similar task to the natural language understanding module of a dialogue system, but usually at a much shallower level.

The document processor indexes the collection of documents from which the answer will be retrieved with information that enables retrieval of paragraphs. Using the output of the question processor, the document processor extracts paragraphs which could answer the user's question. Quite often the processing done at this stage is quite shallow and the only constraints imposed on the paragraphs retrieved are to ensure that they contain all or most of the keywords and at least one word in the same category as the expected answer type (Clarke et al., 2000). However, there are systems which expand the list of query terms returned by the question processor with semantically related terms (Ittycheriah et al., 2000) or implement advanced metrics which measure the plausibility of the paragraph containing the answer (Moldovan et al., 2000).

The answer extraction module takes the paragraphs retrieved by the document processor and locates the actual answer of the question. This is usually achieved using either answer-type pattern extraction or N-gram tiling (Jurafsky and Martin, 2009). The answer-type pattern extraction method relies on handcrafted or automatically extracted patterns to identify the answer to a question. These patterns can be either surface-based or they can rely on syntactic and semantic information extracted by the question processor. The N-gram tiling method uses a large collection of documents such as the Web to extract recurring snippets

which are then scored according to how likely they are to be the answer to the question. For question answering systems which go beyond the factoid QA, the answer extraction stage can require fusing information from several documents in order to generate the answer. From this perspective, this stage is similar to the natural language generation component of dialogue systems. In some cases, an answer validation module is used in order to check the validity of the answer obtained. Magnini et al. (2002) describe answer validation as "filtering out improper candidates by checking how adequate a candidate answer is with respect to a given question". For example, asking "Which president succeeded Jacques Chirac?" the user expects a name of some president as an answer; therefore any answer which is not an instance of president would be incorrect (Grappy and Grau, 2010). Different techniques can be used to check whether the given answer is valid with respect to a question (Rodrigo et al., 2008; Grappy and Grau, 2010; Tonoike et al., 2004; Li et al., 2008).

The vast majority of the existing QA systems are unidirectional pipelines where each module is run in sequence and the user cannot intervene at all. However, Moldovan et al. (2000) investigate the use of automatic feedback loops in cases where there are too many or too few answers available to rewrite the input of the document processor. This is similar to what happens in interactive question answering systems where the system asks for clarification from users.

2.5 Interactive question answering systems

An interactive question answering system is defined as *a QA system that supports at least one exchange between the user and the system, and that allows the user some control over the content that is displayed and the actions that are taken* (Kelly et al., 2009). However, we argue that it is necessary to have more than one exchange in order to consider a QA system interactive. The similarity between dialogue systems and interactive question answering systems is emphasised by the definition given in (Lester et al., 2004):

> *Conversational agents exploit natural language technologies to engage users in text-based information-seeking and task-oriented dialogs for a broad range of applications.*

On the basis of these definitions, we argue that any interactive QA system can be considered a dialogue system, but only those dialogue systems which are information-seeking can be regarded as interactive QA systems. This point of view is adopted in the remainder of the section to decide which approaches to present. An example of output from the system developed by Varges et al. (2007) is presented in Table 2.1. The output in this case is very similar to that of a simple QA system, but the authors argue that in cases where there are more than 10 results, presenting all of them in this format is no longer an option and a dialogue should be initiated to narrow down the request. In contrast, Table 2.2 presents a conversation of the thesis author with an online version of Eliza, a chatbot that behaves like a Rogerian psychotherapist. As can be seen, the chatbot

User:	I want to find an inexpensive Japanese restaurant that takes reservations
Sys:	I found 9 inexpensive Japanese restaurants that take reservations
Sys:	Here are the first few:
Sys:	GINZA JAPANESE RESTAURANT
Sys:	OKI SUSHI CAFE
Sys:	YONA SUSHI
Sys:	Should I continue?

Table 2.1: Example of an output from Varges, Weng, and Pon-Barry(2007)

Eliza:	Hello. My name is Eliza. How may I help you?
User:	I wonder where to have dinner tonight. Do you have any suggestion?
Eliza:	We were discussing you, not me.
User:	Ok, I would like to have Italian food tonight.
Eliza:	Can you elaborate on that?
User:	I like pizza.
Eliza:	Tell me more...

Table 2.2: Example of conversation with the implementation of Eliza from http://nlp-addiction.com/eliza/

does not try to answer the questions, but carries out a conversation and therefore cannot be considered an IQA system.

Interactive question answering systems give users the opportunity to ask questions and refine them on the basis of feedback received from the system until they find the answer needed (Webb and Webber, 2009). Some interactive question answering systems also allow follow-up questions when the user wants to obtain more information related to the question or clarify the answer, so that IQA becomes an iterative process (van Schooten et al., 2009). For this reason, we can consider interactive question answering systems a step forward from simple

question answering systems by offering users a more natural way of finding the information they need. Moreover, given that the sought information is obtained in an iterative manner by asking several simple questions, instead of combining all of them in one complicated question, the IQA systems could be more accurate than usual QA systems. This is due to the fact that IQA systems are less likely to have to deal with ambiguous language constructions such as prepositional phrase attachment or complicated syntactic structures. Moreover, when an IQA system encounters such an ambiguous construction, it can initiate a dialogue to clarify the user request.

As mentioned in the previous section, there are systems which automatically expand or remove some of the query terms used to find potential answers to a question. In a similar manner, one of the most common ways of interacting with IQA systems is by asking users to clarify their questions by rewriting them when the system provides too many answers or too few. From this point of view, interactive question answering should be seen as a mixed-initiative dialogue (dialogue where the initiative passes back and forth between the discourse participants). This is due to the fact that even though the user is the one who normally asks the questions, it is possible for the computer to take the initiative when it requires clarification.

IQA systems give users the possibility to have a dialogue, which makes them similar to dialogue systems; however they differ in several ways in this respect. The interaction between IQA systems and human users is more task-orientated and usually involves fewer turns. Also unlike a dialogue system, they usually

lack "human" characteristics, so interaction with them seems less natural. IQA is orientated towards information seeking, this is why IQA sessions have a clearer structure than dialogue systems and in most cases it is obvious whose turn is next. In both information-seeking dialogue systems and IQA systems the common ground and context can be established by making the system domain-dependent. In this way, both the human participants and the computer can interpret the utterances with respect to that domain and diminish the extent of ambiguities in questions and answers.

2.6 Types of IQA systems

The field of interactive question answering is quite new and there are not many systems available. For this reason, the existing systems differ a lot in terms of their structures and the ways they are implemented. In this section, we will present the most important strategies used for developing IQA systems.

There are two ways of developing IQA systems: by producing a scaled-down version of information-seeking dialogue or by integrating additional functionalities into a standard QA system. These two approaches are presented next. In addition to these two approaches, researchers have also experimented with follow-up questions where the system interacts with users by proposing which questions it can answer next.

2.6.1 IQA as a problem of constraint management

Several systems treat IQA as a problem of constraint management. They identify constraints in the questions and interact with the user when these constraints need to be modified. Usually, most of this processing is done by the dialogue manager. This is the approach taken in our research.

Qu and Green (2002) developed a module which helps handling of under- or over-constrained requests in terms of the dialogue systems. It uses a Constraint-Based Problem-Solver (CBPS) which enables a dialogue system to 1) incrementally interleave query construction with solution construction, 2) immediately detect under-constrained and over-constrained information requests, and 3) provide cooperative responses when these types of problems are detected. The model was implemented in COMIX, a prototype system providing airline flight information. This system queries a relational database of airline flight information using a form-based user interface. With the help of this interface, the user specifies an information need by filling in fields on a query form. If there are some ambiguous values, COMIX takes the initiative and displays a clarification dialogue window. After all fields are filled in, the system submits a query to the database. If the request was over-constrained, using information provided by the CBPS, constraints to relax are suggested.

For the interaction with the user, the system relies on a frame-based dialogue manager which employs a Constraint-Based Problem-Solver (CBPS), consisting of three subcomponents: *Solution Construction*, *Solution Evaluation* and *Solution*

Modification. The Solution Construction module collects constraints and queries the database to get results, therefore fulfilling the role of answering the question. The Solution Evaluation module evaluates the results to determine whether a satisfactory solution has been found. If it determines that the query is currently over-constrained or under-constrained, the Solution Modification module studies constraints and tries to identify relaxation or restriction candidates which are presented to the user.

Results of evaluation showed that the CBPS helps to enhance the performance in terms of dialogue efficiency and also improves the success scores for the task completion. This was expected given the presence of over-constrained queries in these tasks. The CBPS gave an opportunity to offer cooperative relaxation suggestions to make dialogue more effective.

Varges et al. (2007) explored the ways a dialogue system managed the results of database queries phrased in natural language. Their aim was to find efficient ways of managing a dialogue and providing a sufficient amount of information to users so that they are neither overwhelmed with too much information, nor left uncertain about some details. The authors describe several systems dealing with restaurant selection, MP3 player operation and navigation tasks. Their goal is to choose a single item out of a larger set of items and at the same time make the dialogue as natural as possible. The task of interactive question answering is presented in the context of a much larger system that also includes speech language understanding and text to speech generation modules.

The interaction with the user is coordinated by a dialogue manager which uses

	# results	Modified	System's answer
(1)	Small	No	There are 2 cheap Thai restaurants in Lincoln in my database: Thai Mee Choke and Noodle House.
(2)	0	No	I'm sorry but I found no restaurants on Mayfield Road that serve Mediterranean food
(3)	Small	Yes	I found no cheap Greek restaurants that have a formal dress code but there are 4 inexpensive restaurants that serve other Mediterranean food and have a formal dress code in my database: ...
(4)	Medium	No	I found 9 restaurants with a two star rating and a formal dress code that are open for dinner and serve French food. Here are the first ones: ... .
(5)	Large	Yes	I found no [NP-original]. However, there are N [NP-optimized]. Would you like to try searching by [Constraint]?

Table 2.3: Strategies for dealing with user questions

a content optimisation module and an ontology of constraints. The role of the optimisation module is to control the amount of content, resolve ambiguities and provide recommendations to users. The ontology contains information about three major types of constraints: hierarchical, linear-ordinal and binary. Depending on the type of constraint and number of results returned by the system, different strategies of constraint relaxation are used. Dialogue strategies for dealing with query results are manually built and thresholds are predefined. Table 2.3 presents some examples extracted from (Varges et al., 2007) showing different dialogue strategies for dealing with user questions.

Table 2.3 presents the output of the system when different questions are asked. It also indicates the number of results available (the *# results* column) and whether a modified list of the constraints is available (the *Modified* column). In the first example, the system finds a small number of answers to the question and returns all of them without suggesting any modifications. The system treats the question in example (4) in a similar manner, but does not display all the results because there are too many. Example (2) shows how a simple QA system answers when there is no answer to a question. An answer can be obtained by modifying, automatically or interactively, the constraints of the initial query as shown in example (3). Example (5) shows how an IQA system initiates a dialogue for narrowing the number of answers returned to a user, by clarifying the user request.

The authors perform two evaluations: a general evaluation and a controlled experiment to test the use of a suggestion strategy. General tests achieve a 94.44 % completion rate of the given tasks and overall user satisfaction with the system results. The controlled experiment shows that users prefer to get suggestions only when they have too many matches, but it is not crucial in cases where there are no matches.

Rieser and Lemon (2009) use an approach similar to the previous one and adapt dialogue policies for QA to obtain an IQA system. They work with a domain of in-car and in-home applications and provide examples of dialogues for choosing a song for a playlist. The main focus of their research is how to acquire enough reliable query constraints, how many results to present to a user and when to present them. They try to solve the problem of trade-off between the

length of the dialogue (how many questions are asked) and the number of items in the resulting list. Their system learns the best dialogue strategies for interactive question answering, making it possible to choose the most appropriate dialogue move. Evaluation of the system showed significantly better results of the learnt policies compared to the hand-coded threshold-based policy. In 93 % of the cases the learned policies outperformed the hand-coded ones. In contrast with the IQA systems presented above, this system is meant to be open domain, trying to answer any question.

2.6.2 Enhanced QA

Magnini et al. (2009) rely on an enhanced QA system to develop an IQA system. In contrast to the approaches described in the previous section, they use a traditional QA system as a starting point and suggest ways of interacting with the user in cases of failure. Table 2.4 presents some examples of interaction with the system.

The authors rely on a special IQA ontology as an addition to the domain ontology. This ontology captures significant aspects of interaction and becomes a basis for analysis of both questions and answers, and is used in modelling dialogue templates. They analyse a question according to three features: *ExpectedAnswerType (EAT)*, *Constraints* and *ContextOfUtterance*. *EAT* is the semantic category associated with the desired answer, *Constraints* help to specify and restrict the information sought as an answer to a question and *ContextOfUtterance* adds additional information to the question such as the time and place where the question was uttered. The ontology also dictates

Question:	Where is Titanic by Spielberg showing today in Trento?
Answer:	I don't know of any movies named Titanic directed by Spielberg. Titanic by Cameron is showing at Cinema Astra.
Question:	Where is Titanic by Spielberg showing today in Trento?
Answer:	I don't know of any movie Titanic directed by Spielberg. Are you interested in Titanic by Cameron or in movies by Spielberg?
Question:	Where is Shrek showing in Trento?
Answer:	No theatre is showing the movie Shrek today. Are you interested in tomorrow's scheduling?

Table 2.4: Interactions with the system in case of failure

the components of the answers: *CoreInformation (CoreInfo)*, *Justification*, and *ComplementaryInfo (ComplInfo)*. *CoreInfo* represents the actual answer to the question, *Justification* gives some contextualisation of the answer and *ComplInfo* is additional non-mandatory information which can be added to any type of answer if considered useful. In cases where a failure is encountered, the reason for this failure is analysed with the help of the ontology and dialogue templates are used to create a natural dialogue.

Dornescu and Orăsan (2010) show how it is possible to implement interaction in the QALL-ME framework (Sacaleanu et al., 2008). The QALL-ME framework is an architecture for implementing closed domain QA systems. At the core of the QALL-ME framework there is a domain ontology and a text entailment engine, which are used, together with other language processing components, to generate

procedures to answer questions. In the case of the prototypes developed in the QALL-ME project these answering procedures are in SPARQL, a query language that extracts data from an RDF (Resource Description Framework) database. Dornescu and Orăsan (2010) propose a solution which inserts metadata indicating the system's understanding of the question with respect to the underlying domain ontology in the SPARQL queries returned. This metadata contains both information about the expected answer type and the constraints associated with the question. When the system fails to answer a question, it uses the metadata to begin an interaction with the user asking which constraints to modify. In this respect, the solution proposed by them is similar to the systems presented in the previous section.

Quarteroni and Manandhar (2009) present an IQA system, YourQA, which combines an open-domain QA system with a chatbot. The underlying QA system is open domain and relies on the Web to obtain answers to both factoid and non-factoid questions. For this reason, it can be argued that it is a question answering system enhanced by dialogue interface. The authors opted for the use of AIML (Artificial Intelligence Markup Language) to add some interactivity. The system is based on pattern matching; the last user utterance is matched against a range of dialogue patterns and then a coherent answer is produced using a list of responses associated with such a pattern. Several evaluations were performed and showed that most users (in the first evaluation - 87.5 %, second - 58.3%) prefer the interactive interface of the system instead of the simple QA system.

Liu et al. (2010) discuss the benefits of using IQA for dealing with the domain of

computer fault diagnosis. They address the problem of IQA by using the domain ontology and answers that have already been formulated. The initial question analysis maps questions to some basic concepts that can be found in the ontology; if not enough or too many concepts are found, the interactive strategy is used to deal with this situation. The paper reports the increase of the precision from 13% for simple question answering to 81% for IQA.

Tang et al. (2011) deal with the situations where questions are unclear and unspecific: additional questions are generated to make users aware of ambiguities and suggest some refinement of the question. Chinese WordNet is used to generate these additional questions. Several examples of this approach can be found in the paper, e.g. the question "Which hospital is better?" can trigger questions such as "Which city? Shanghai, Suzhou, ..., or Shenzhen?". The paper reports the precision of recommendations; however the overall impact of the QA is not evaluated.

2.6.3 Follow-up questions

Question Answering systems that can deal effectively with follow-up questions (FQs) can also be considered to belong to the domain of IQA. van Schooten et al. (2009) tackle the problem of follow-up question classification and discuss challenges which need to be addressed while handling follow-up questions. They identify four main types of follow-up questions: self-contained questions, regular FQs (need dialogue context information to be understood), discourse questions and negative feedback (or meta-questions). Each kind of question has to be treated in a different

way; therefore at first the type of question has to be identified correctly. The authors focus on the stage of context competition which is vital for the successful work of QA systems offering processing of FQs. Context completion consists of 3 steps: identification of need for context completion, identification of rewriting strategy and anaphors and referent selection. Each of these stages introduces its own challenges which need to be addressed; for example, topic shifts sometimes make it difficult to classify FQs correctly. Using examples of two systems (IMIX (Boves and den Os, 2005) and Ritel (Galibert et al., 2005)), the authors show how this approach can bring interaction to classical QA.

Harabagiu et al. (2005) try to predict the range of questions a user is likely to ask given a context and a topic, in this way producing an interactive system. After asking a question, a user gets not only the answer, but also suggestions for some follow-up questions. This is done by using a special database of "predicted" questions and answers (QUAB). Their IQA system (FERRET) uses the similarity between the user question and questions in QUAB to suggest follow-up questions. Users can ignore the suggestion or select one of the proposed questions. The system is evaluated on questions about weapons of mass destruction. The evaluations show that usage of QUAB helps to improve both efficiency and effectiveness of the system and contributes to the rise of user satisfaction. However, it also revealed that the system is still not very accurate.

Bernardi and Kirschner (2008), Kirschner and Bernardi (2009), Kirschner et al. (2009) and Bernardi et al. (2010) study the way answering follow-up questions can benefit from the context. Bernardi and Kirschner (2008) claim that an IQA system

can be improved by predicting the focus of the follow-up questions. They study what makes a dialogue coherent, what "things" users focus on and how this focus evolves. Based on these theoretic findings, they train a module to track the focus in dialogues and show that it helps to improve the performance of an IQA system. This research is continued in (Kirschner and Bernardi, 2009) which studies follow-up questions that are not topic shifts, but rather continuations of the previous topic. The authors identify different relations which can hold between follow-up questions and preceding dialogue and show how this model can help to select the correct answer among several answer candidates.

Using the previous results, Kirschner et al. (2009) explore ways in which the distinction between topic shift and continuations of the previous topic can improve the results of IQA systems. They notice that it is crucial when deciding whether or not to apply context fusion techniques for retrieving the answer. They rely on shallow features such as lexical similarity, distributional similarity, semantic similarity and action sequence. Even though the use of shallow features reports some promising results, Bernardi et al. (2010) enhance their previous model by adding deep features based on discourse structure and centering theory. However, the results showed that these features do not outperform the shallow ones on their own, but that a combination of shallow and deep features increases the precision of an interactive question answering system.

Schooten and Akker (2011) also regard IQA mostly as a QA system handling follow-up utterances of the users. Once the users are presented with an answer or are informed of the system's failure to find an appropriate one, naturally they

may be interested in asking more questions. The main aim of the system then is to decide what kind of follow-up question it encountered and what strategy should be used. Their research explores different context completion strategies that allow the system to pass context to the QA for answering follow-up questions. Several basic approaches can be used: 1) questions can be rewritten as self-contained questions, 2) the query of a follow-up question is combined with that of previous utterances, 3) search can be done within previous results. The first strategy can be used when the follow-up question contains ambiguities, anaphoric expressions and elliptical sentences. The second one allows for combination of information that is already available with some that has been previously acquired, resultong in a more complete query. Another option is to use previously acquired search results as a corpus for a follow-up question search. This strategy of incremental refinement allows users to ask several simple questions rather than one complicated one. However, in order to use all the described strategies, follow-up questions should first be classified according to their type; for this, corpora of follow-up questions are used.

Studies of the context of the question can be regarded as similar to those of follow-up questions. Several recent papers (Wang, 2011; Waltinger et al., 2012) study the ways in which the type of the question and the context information it contains can be used for interactive QA. Waltinger et al. (2012) study not only question classification, which is often used in classical QA, but also address the problem of the focus and topic of the question. They try to identify the main focus of the question and also information that helps to specify it. Contextualisation -

identification of additional contextual information, of the questions proves to be beneficial for question answering systems and enhances their application within interactive scenarios in terms of both response time and correctness.

2.7 Evaluation

As in many other fields in computational linguistics, evaluation plays an important role in IQA. Despite this, there are no methods that are specific to the field and the only existing options are adaptations of the evaluation methods used in QA and dialogue systems. The main difficulty of designing an evaluation method for IQA lies in the fact that it is rarely possible to predict in advance how an interactive session will evolve. For this reason, it is necessary to have humans involved in the evaluation process, making the process slow, expensive and difficult to replicate. Given that there is no framework for IQA evaluation, this section focuses on the evaluation methods used for QA and dialogue systems applied to IQA systems.

QA evaluations differ depending on whether they evaluate factoid or complex questions (such as definition, relationship and scenario questions). One way of evaluating is to create a set of "gold standard" questions and answers and see how successfully a system matches this gold standard. This method is not very robust for complex questions where the correct answer can be expressed in many different ways or there can even be several possible answers. In this case human-in-the-loop evaluations (like TREC[1] evaluation competitions) are used, where human assessors are involved in the process. The above-mentioned techniques can also be used for

[1]Text REtrieval Conference (TREC): http://trec.nist.gov/

IQA answering to check the correctness of the answers returned to users; however on their own they do not provide enough information about the quality of an IQA system. For this reason, evaluation methods for dialogue systems are used as well.

The QA-CLEF 2008 evaluation competition tried to simulate a pseudo-interactive QA session by presenting questions grouped into topics (Forner et al., 2008). Some of these questions contained coreferences between each other, and there were also questions which relied on the answer to a previous question. Despite arranging this pseudo-interactive setting, each question was assessed individually, not like in a real interactive session.

Evaluation of dialogue systems is also a tricky task and a lot of different methods of evaluation can be found in the literature. Dale et al. (2000) suggest using task-based evaluation where quality of a dialogue system is measured in terms of the task completion. The same approach can be found in (Jurafsky and Martin, 2009) where the authors suggest measuring three different characteristics:

- completion success (how correct the solution is observed to be when using the dialogue system);
- efficiency cost (how much time is spent on finding a solution, for example, number of turns);
- quality costs (how well the system works, for example, number of times the user has to interrupt the system).

Spitters et al. (2009) address the last characteristic by simply asking people to complete a questionnaire and rank the quality of the system by giving grades.

These questions include, for example, a request to evaluate the naturalness of the system in general, but the objectivity of such a method is debatable.

Harabagiu et al. (2005) note that in terms of IQA (and in dialogue systems) dialogue as a whole is usually evaluated in terms of:

- efficiency, defined as the number of questions that the user must pose to find particular information;
- effectiveness, defined by the relevance of the answers returned;
- user satisfaction.

Given that questionnaires are used to evaluate dialogue systems(DS), it seems natural to adapt the same strategy for IQA, which will provide an opportunity to evaluate various aspects of the system's performance.

Dornescu and Orăsan (2010) ask human assessors to rank alternative questions generated by the system in cases where the system cannot answer a question. In this way, they try to find out which constraints users prefer to be modified first in order to obtain an answer.

Kelly et al. (2009) describe a first attempt to combine methods from QA and DS to develop a general framework for IQA evaluation. This framework relies on adaption of the existing technologies or creation of new ones. The authors use three questionnaires (Cognitive Workload, Task and System Questionnaires) and adapt them for specific IQA applications. This method provides ways to evaluate systems from different angles, but involves a lot of preliminary manual work and time-consuming labour on the part of the human assessors.

Sun et al. (2011) offer a novel method that can be used for evaluation of IQA systems. They introduce the X-EVAL model which consists of two stages: using the experimental systems and creating some type of final work at the first stage and evaluating each other's work at the second stage. This method was previously used for the evaluation of collaborative IR. The goal of the research is not to compare several systems, but rather to prove that the suggested methodology is capable of distinguishing among several systems. The experiment was organised in a specific way, so as to eliminate all the factors which can affect the objective evaluation of the system. Therefore, analysts were asked to work with different systems and different scenarios and produce several reports. After this, each analyst was asked to evaluate reports prepared by others according to seven criteria, including presence or absence of redundant information and good organisation of the material. The rotation of the analysts, scenarios and systems ensured that no factors interfered with the evaluation of the effectiveness of the systems. The experiment showed that the X-EVAL model works well for IQA as well and can indicate the impact of a given system on the completion of the task.

As in many other application-orientated fields where complex systems need to be evaluated not only as end-to-end systems, evaluation of IQA systems can be also performed at component level. This entails applying the relevant evaluation methods to each component on its own. However, these evaluation methods are component-specific and are beyond the scope of this chapter.

2.8 Conclusions

This chapter provided background in Interactive Question Answering which is the main focus of this research. It also emphasised the potential of IQA for addressing the problem of intuitive and effective search for information while highlighting the lack of research in this field. However, a review of existing work on Information Extraction, with emphasis on coreference resolution and relation extraction, was also carried out. The findings of this review are presented in Chapters 4 and 5.

Chapter 3

Data preparation and its characteristics

This chapter describes the data characteristics and presents the steps undertaken to prepare it for the experiments described in the chapters to follow. First, we describe types of texts used for our research and the process of corpus collection (Section 3.1). We then proceed to discuss the extraction of features describing mobile phones in Section 3.2.

3.1 Types of texts

As previously discussed (Chapter 1), we aim to retrieve information from natural language texts using NLP techniques. We also believe that NLP can be helpful when building IQA systems by limiting human effort, making the process more objective and less time-consuming. Therefore, we need to create a corpus of texts that will contain relevant information and be representative enough of our domain. In light of this, two types of texts are used in this research: semi-structured texts from Wikipedia (described in Section 3.1.1) and unstructured texts from the domain of products (discussed in Section 3.1.2).

3.1.1 Wikipedia

One of the corpora compiled for this research consists of Wikipedia texts describing mobile phones. Wikipedia is a semi-structured source of information that is collaboratively built by many users. It is a multilingual, web-based, free-content resource and is updated on a regular basis by largely anonymous Internet volunteers. Every user can correct content added by others to ensure that the final result is up-to-date and reliable. The choice of this resource was motivated by several factors: it is characterised by a pre-defined structure and is more homogenous and reliable than ordinary Web data; and it has some additional features that can be regarded as very promising for information extraction - for example, the presence of links can assist the identification of Named Entities and facilitate the task of information extraction.

In our research, the Wikipedia corpus is used to extract features of mobile phones (Section 3.2) and is also employed using the methods described in Chapters 4 and 6.

When building our corpus, the first problem we faced was finding ways to automatically extract texts describing different mobile phones and delimit them from the texts belonging to other domains. The solution adopted is presented in the next section.

3.1.1.1 Corpus building

One of the features of Wikipedia articles is the possibility of assigning them to different categories. Consequently, when creating a new article, the authors can

indicate a category/categories they find relevant to describe the article. Therefore, the most intuitive way to distinguish the domain of mobile phones from other domains is to select all articles which have a relevant category assigned. For example, if interested in articles describing mobile phones, we would choose the pages tagged with the category "mobile phones". However, the creators of Wikipedia articles are not limited to choosing only one category and can use multiple ones. These categories can also be of different level of generality, which makes the approach of selecting articles described by a specific label problematic.

Wikipedia categories are not a taxonomy, but a folksonomy, a system of classification derived from the practice and method of collaboratively creating and managing tags to annotate and categorise content (Peters and Becker, 2009). Folksonomies allow large disparate groups of users to collaboratively label massive and dynamic information sources. However, even though the benefits of this approach are obvious, there are a number of disadvantages that have to be taken into account when working with folksonomies. Categories are assigned by the authors of the page and so can be subjective because every person can have a different perception of what the page is about. Folksonomies may also suffer from inconsistencies, for example, misspelled tags, and different variants of tag names. This type of classification usually lacks hierarchy as well, which makes it difficult to find links between similar/related categories or to search using them. All this results in folksonomies being less reliable than classical taxonomies. All of these characteristics of the Wikipedia categories organisation had to be taken into account in our research.

It turned out to be not possible to choose a general category, for example, "mobile phones", and obtain all the articles to which it is assigned. Using this approach, the number of texts is too small and they are not representative of the domain. Figure 3.1 presents the category tree for "mobile phones" and for each category provides statistics on the number of subcategories and pages assigned to it. This figure reveals that only 62 pages in Wikipedia are directly assigned the category "Mobile phones", but this category itself has 16 subcategories. The low number of articles assigned to "mobile phones" can be attributed to the fact that general categories are rarely assigned to many articles, otherwise the list of categories for each article would be too extensive. Another option to find relevant articles for our domain is to select a very specific category, but then the problem of coverage will arise anew: it will be too specific and will describe only a small amount of articles, for example, the category "Watch phones" is assigned to only six pages.

However, some categories are organised in hierarchies; therefore, at times, links between general categories and more specific ones can be found. Using this knowledge, we can attempt to combine information about both types of categories: a general category (it will help to delimit the domain) and all its subcategories (it will address the problem of coverage).

The category "Mobile phones" was chosen as the most descriptive category for our domain; as mentioned before, it was assigned to only 62 pages but had 16 subcategories. All the pages which are assigned this category or any of its subcategories were extracted from Wikipedia, which resulted in a total of 1800 files.

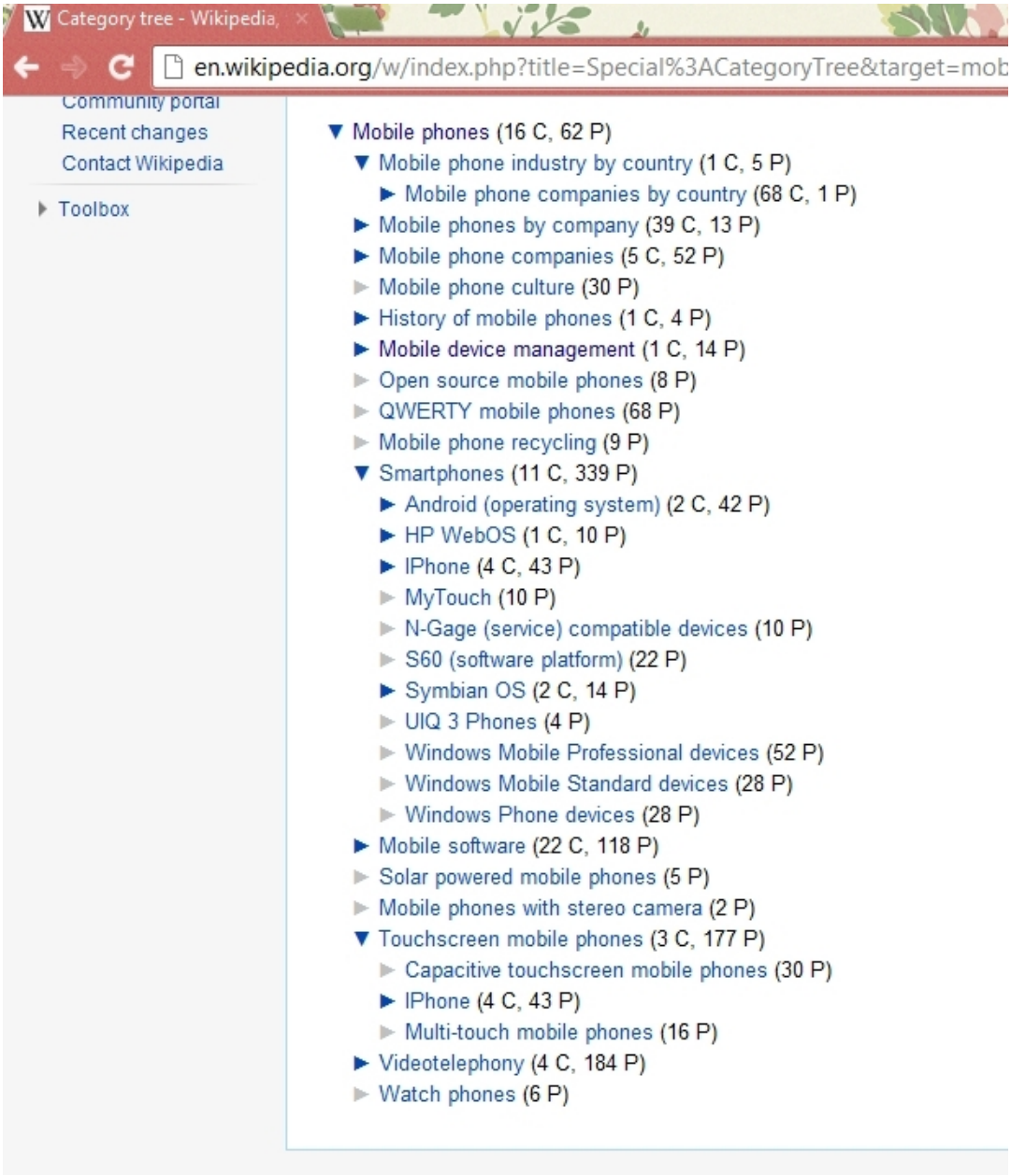

Figure 3.1: Category tree for Wikipedia category “mobile phones”.

However, further analysis of the extracted texts revealed that many of the articles did not describe mobile phones, but rather various aspects related to mobile phones such as companies, software, games, etc. It indicated that only using categories does not yield desirable results.

However, most Wikipedia articles also feature infoboxes, brief tabular information about the article. Wikipedia's help page[1] describes infobox as "a fixed-format table designed to be added to the top right-hand corner of articles to consistently present a summary of some unifying aspect that the articles share and sometimes to improve navigation to other interrelated articles". Infoboxes appear on the top right hand side of the articles pages and quickly summarise important points in an easy-to-read format. The initial idea is that the infobox is introductory for the article and contains, primarily, material that is expanded on and supported by citations to reliable sources elsewhere in the article. Infobox templates describe the type of information which is common to related articles. For example, articles about phones would contain information about dimensions of the phone, its weight, etc. Infobox templates are classified and have types guiding what kind of information should be added there; therefore, the type of infobox used can be treated as an additional way of classifying Wikipedia articles. Taking into account this information, only articles with infobox type "mobile phones" were chosen as a corpus for investigation. This strategy helped to obtain a total of 560 articles (216,555 words) which was deemed enough for this research, and also guaranteed that all of them belonged to the domain of mobile phones.

[1] http://en.wikipedia.org/wiki/Help:Infobox

All Wikipedia articles come with a special markup. Some markup such as headlines, links to other pages, italics, etc. can provide valuable clues for our Information Extraction task and can be helpful for the identification of Named Entities as all these features show where the boundaries of Named Entities are. However, at this stage of our research, this markup was ignored.

As previously mentioned, most of the articles in Wikipedia have infoboxes that can later be used for the extraction of product features and their values. Even though information in terms of the same type of infobox is not always standardised, it seems promising to use them as an initial step of semi-automatic extraction of product characteristics from the texts. This idea is further developed in Section 3.2.

3.1.2 Unstructured texts

As mentioned before, we collected two corpora and one of them contained general, unstructured texts. One of the hypotheses of our research was that the use of opinionated texts can be beneficial for the ranking of the phone features. Therefore, texts featuring customer reviews about different mobile phones were collected. In addition, the choice of the user reviews allowed us to build a homogenous corpus that discusses the topic of interest – different models of mobile phones – and at the same time each text (user review) is focusing only on one phone. This makes this corpus similar to that of the Wikipedia articles, which is important for the success of our research.

The corpus of reviews is less formal than Wikipedia articles and provides

human opinions about some concrete models of the phone. Our hypothesis is that this subjective information can be employed for the ranking of the features and identifying which features are of a bigger interest for the users. More details about this step are provided in Chapter 6.

3.1.2.1 Corpus description

For our experiments, we compiled a corpus of reviews from the Epinions.com[2] website, which provides customer reviews about different categories of products. This website has a specially developed system that encourages users to write high-quality reviews, and therefore it seems more reliable than a well-known analogous service from Amazon. Additionally, these reviews tend to be longer, more detailed and cover more aspects of the product than the ones found on the Amazon website. All the texts differ in size and were written by different contributors.

Since our research focuses on the domain of mobile phones, we collected reviews from the category *Cellular Phones* on the 21st October 2011. Classification of the reviews is much simpler than the system of categories used in Wikipedia. Therefore, it was easy to decide which reviews were relevant to a topic of interest. "Cellular phones" was one of the subcategories of the "electronic products". All the articles belonging to that category were included in our corpus. The acquired corpus is composed of 3,392 reviews (114,708 sentences; 2,253,877 words) and all the texts are organised under two labels: "yes" and "no". These labels reflect the users' opinions about the product and whether they would recommend it or not,

[2]http://www.epinions.com/

overall. We currently have 2,437 reviews with the label "yes" and 955 with the label "no". However, at this point of our research, we are not using the general label of the text; therefore a creation of a balanced yes/no corpus is not relevant to this thesis. The main interest of our research is to find opinionated sentences describing mobile phones and their features and each review, disregarding the general label it has, usually provides both positive and negative comments about some model. The inclusion of all the articles from a chosen category (in our case "cellular phones") ensures that all the texts describe various mobile phones. A corpus for a different domain can easily be built using the same steps, but starting from a different relevant category.

Each review in the corpus is represented both as a text and as an xml document that contains not only the text of the review but also some additional information. It features the user name of the reviewer, the time the review was submitted, and the name and the brand of the product reviewed, as well as the star rating. This additional information is not currently used in our research but may be interesting for additional investigations in the future.

The two corpora presented above are used throughout this thesis. Firstly, the Wikipedia corpus is used to extract features of mobile phones and this research is presented in the next section. This corpus is also employed using the methods described in Chapters 4 and 6. The review corpus is used with the methods discussed in Chapters 4, 5 and 6.

3.2 Features

The focus of our research is to create a natural dialogue to assist users in choosing a product, in our case a mobile phone. As mentioned before, all products can be described in terms of a set of features and their values. For example, a mobile phone can have a a feature "camera", which can take different values like "5 mpx", "8.0 megapixels", "with zoom". Combination of features and their corresponding values gives a description of a concrete phone model. Therefore in order to create an IQA system for helping to choose a product, we first need to identify the product features to be used for interaction with the users. Creating such a database completely manually can be subjective, time-consuming and labour-intensive. This is why finding semi-automatic ways to get the needed information is very important for our research. We explored possibilities of using corpus-based methods which rely on ngrams and terminology extraction, this research is described in Section 3.2.1. Section 3.2.2 discusses the possibility of using the existing resources for this task.

3.2.1 Corpus-based methods

In order to find the phone features, we started by analysing our corpus. We were interested in discovering what the characteristics of the corpus were and whether they could be subsequently used to achieve our goals. We attempted to use two main corpus-based approaches: ngrams and terminology extraction methods, which are described in the following sections.

3.2.1.1 Ngrams

At first we attempted collecting ngrams based on our corpus of Wikipedia articles describing mobile phones (Section 3.1.1). We assumed that, given the texts describe different phones, lists of ngrams sorted by frequency can assist us in identifying the most important features of a phone. The initial pre-processing was used to clean the texts from the additional Wikipedia specific markup and to tokenise them. Then, different ngram lists were collected: bigrams, trigrams and 4-grams.

Table 3.1 shows the list of ngrams that were extracted from our corpus. It reveals that several problems were encountered when using just a simple ngram approach. One of its major drawbacks is not taking into account linguistic information and splitting the units in such a way that the resulting ngram is meaningless. For example, we are more interested to see NPs and VPs, rather than sequences like "used to charge", "locations or when", "Latin alphabets are" which would be considered meaningless for the information extraction task. The second drawback is the presence of function words like articles and prepositions. This problem should be also addressed in order to get high-quality information.

Table 3.1 reveals that bigrams were not very informative for our needs: they contain a lot of words of general vocabulary, which cannot be considered domain specific and thus help to find features. After examining the list of the top bigrams collected from the corpus, it becomes obvious that it would greatly benefit from the addition of POS tags to filter out such entities as "it was", "with a", etc.

Freq	Bigrams	Freq	Trigrams	Freq	4-grams
998	the phone	**158**	of the phone	**63**	is a mobile phone
920	of the	**126**	the phone is	**41**	in the united states
600	up to	**102**	the phone has	**34**	the phone has a
582	is a	**95**	as well as	**29**	also known as the
499	in the	**80**	version of the	**27**	fm radio with rds
446	on the	**75**	a mobile phone	**27**	it is possible to
374	to the	**71**	memory card slot	**23**	it is available in
353	it is	**71**	one of the	**23**	side of the phone
254	the nokia	**69**	is a mobile	**23**	the phone has been
252	can be	**69**	was released in	**21**	multiple numbers per name
250	with a	**63**	is available in	**21**	of the phone is
239	with the	**56**	also known as	**21**	there is also a
234	mobile phone	**55**	known as the	**21**	up to 2 gb
231	and the	**54**	it has a	**19**	240 x 320 pixels
226	has a	**54**	megapixel camera with	**19**	a 2 megapixel camera
220	as the	**51**	the phone was	**19**	mobile phone manufactured by
218	and a	**49**	in the united	**18**	in addition to the
209	for the	**49**	the phone can	**18**	is one of the
202	is the	**47**	the phone also	**18**	it was released in
179	it was	**47**	the united states	**18**	the phone features a

Table 3.1: Top 20 ngrams with their corresponding frequencies

Examining Table 3.1 further, we can observe that trigrams and 4-grams contain more useful information than bigrams. It seems reasonable to attempt to filter out more information like prepositions, articles, etc., so that we can obtain more meaningful output. However, to achieve this, POS tagging is needed and therefore the problem shifts from a simple surface-based analysis to a more linguistically-motivated approach.

To discover whether the addition of linguistic information could help for our task, we parsed the whole corpus using the Connexor parser Tapanainen and Järvinen (1997). Our intuition was that in most cases phone features are noun phrases. However, parser can sometimes mistakenly extract very long NPs. Therefore, it was decided to examine only the heads of NPs in order to capture more precise information with less noise. This approach helped us to filter out some irrelevant information, but it had one serious drawback: even though examining only the heads of NPs helps to remove noise, we miss a lot of features which consist of two or three words. Furthermore, when using the parser, we discovered a range of limitations: the parser was not always able to set the correct boundaries of NPs. In addition to aforementioned extraction of too long NPs that are too long, Connexor did not perform well in cases where specialised vocabulary was used and was extracting NPs that were too short. For example, extraction of features such as "3g network speed" would be problematic because the first part of it, "3g", is considered an abbreviation. Consequently, the use of just the heads of NPs allowed us to filter noisy items, although we realised that the use of the Connexor parser imposes serious limitations and its output should be post-processed to get more

reliable results. More detailed information about the evaluation of all methods is presented in Section 3.2.1.3.

3.2.1.2 Term extraction

Since using ngrams did not prove very promising, it was decided to use terminology extraction methods and study whether they can be helpful for our task. Two different methods of terminology extraction were employed:

1. extraction based on POS tagging and statistics;
2. web-based terminology extraction.

Python library **topia.termextract 1.1.0**[3] was used for the first approach. This library allows to POS tag the texts and extract the most important terms based on these tags. This method relies on simple heuristics to extract noun phrases. It also provides an option to choose the frequency of the extracted terms depending on whether we want to extract terms that appeared only once or are more interested in something that appeared several times. By default, the algorithm will consider extracting a one word sequence as a term if it appeared at least three times in the text. However, multi-word nouns receive "strength", which equals to number of words in the term. If the strength of the term is larger than 1, by default, it will be extracted regardless of the number of occurrences.

We used this terminology extraction tool in order to get a frequency-based sorted list of terms. Two different lists were compiled: one showed frequency of the term in the whole corpus, while the other only showed how many documents

[3]http://pypi.python.org/pypi/topia.termextract/

Freq	**TE-corp**	**Freq**	**TE-doc**	**Freq**	**Yahoo!**
1935	phone	**282**	phone	**136**	nokia
885	nokia	**111**	video	**127**	mobile phone
500	video	**106**	nokia	**64**	sony ericson
449	feature	**104**	feature	**60**	samsung
406	camera	**98**	camera	**37**	verizon
389	gsm	**69**	gsm	**35**	windows mobile
364	samsung	**69**	fm radio	**34**	motorola
327	version	**66**	memory	**32**	verizon wireless
268	screen	**65**	music player	**31**	megapixel camera
254	memory	**64**	samsung	**30**	qwerty keyboard
223	music	**57**	screen	**24**	htc
220	support	**52**	time	**22**	hsdpa
218	lg	**51**	music	**22**	umts
206	device	**50**	support	**20**	lg electronics
199	time	**48**	qwerty keyboard	**18**	music player
198	user	**48**	talk time	**17**	bluetooth
195	motorola	**47**	version	**17**	t-mobile
187	htc	**45**	united states	**16**	mobile phones
186	button	**43**	battery life	**14**	wcdma
181	blackberry	**43**	2 gb	**13**	samsung electronics

Table 3.2: Top 20 terms extracted using terminology extraction methods: **TE-corp** - frequency of a term in the whole corpus, **TE-doc** - how many documents contained this term, **Yahoo!** - how many documents contained terms extracted by Content Analysis by Yahoo!

contained the given term at least once. Table 3.2 shows the 20 top items extracted using this tool. However, it should be mentioned that we had to clean the initial list and filter out several kinds of items: all terms containing punctuation marks (e.g. “), ”, “;mhz”), terms which are letter tokens (e.g. “k”, “e”, etc.) or numbers (e.g.“8”). The overall quality of the extracted information was quite good and provided us with an extensive list of features and their values. More details of the evaluation of the method are provided in Section 3.2.1.3.

We also tried using web-based interface **Content Analysis by Yahoo!**[4] for extracting terminology from our corpus. This service allows to identify concepts/terms inside the text as well as link them to a relevant page in Wikipedia. It is also possible to rank the acquired terms based on their importance for the document and a special score is calculated for each document. This method is similar to the extraction of key terms, which is frequently used in information retrieval for the indexing of the documents. The quality of the lists was higher than those produced by other methods and contained less noise; nevertheless, it needed a lot of adaptation to be used for our task. For example, the results of this approach favoured company names and product names, which, although useful, prevented, the method from extracting other product features.

We can identify the items that are used in all the texts of our collection and then using combined statistics from all the texts filter the ones that are the most frequent. However, when using this approach, we may miss out some terms that do not have high frequency in terms of each document but can be encountered in all

[4]http://developer.yahoo.com/search/content/V2/contentAnalysis.html

documents of our collection. Such items are of interest for our task and thus should be identified. Therefore it was decided to ignore the lists taking into account ranks and just to check how many documents featured the given term. Table 3.2 shows the top 20 items of the frequency-based list extracted using Content Analysis by Yahoo!

3.2.1.3 Evaluation

In order to compare the lists described in the previous sections, an objective evaluation was carried out and the accuracy of each list was checked. We measured only the precision of the methods described above and did not study the recall, which is common practice for terminology extraction (Maynard, 2000). The first 300 items were taken from each list and seven different lists were used:

1. bigrams;
2. trigrams;
3. 4-grams;
4. heads of NPs;
5. results of terminology extraction - frequency of a term in the whole corpus;
6. results of terminology extraction - frequency based on how many documents contained the given term;
7. frequency based on how many documents contained terms extracted by Content Analysis by Yahoo!.

At the next stage, all items were checked manually and those items that did not seem relevant to our task were filtered out. The task was completed by a

No	List name	Relevant items	Accuracy
1	**Trigrams**	70	23.3%
2	**4-grams**	83	27.7%
3	**Bigrams**	89	29.7%
4	**Heads of Nps**	98	32.7%
5	**TE-corp**	154	51.3%
6	**TE-doc**	202	67.3%
7	**Yahoo!**	**216**	**72.0 %**

Table 3.3: Accuracy of corpus-based methods for feature extraction

linguist who was given instructions and informed about the purpose of the feature extraction. Table 3.3 shows the results acquired; for each list we show how many relevant items it contained as well as the accuracy of each method.

It was observed that sometimes it is quite difficult to decide whether an item in the list is a feature or value of the feature. Sometimes, items can be meaningless without context which makes the classification task much harder. Therefore it was decided to carry out an inter-annotator agreement study to get an objective estimation of how difficult the annotation task in question is. This study is described in the next section.

3.2.1.4 Inter-annotator agreement

To study inter-annotator agreement, two annotators were asked to do the same task and annotate relevant items in the lists generated using previously described methods. We calculated kappa agreement between annotators for each of the lists.

Cohen's kappa coefficient is a statistical measure used for assessing inter-

annotator agreement (Carletta, 1996). This measure uses the number of cases that were agreed or disagreed by the rates but also takes into account the hypothetical probability of chance agreement based on the probabilities of each observer randomly selecting each category. It take values in a range of 0 to 1, where value "0" indicates no agreement among the raters other than what would be expected by chance and value "1" signifies complete agreement among the annotators.

First, kappa was calculated separately for each list, the results of which can be seen in Table 3.4. At the next step we calculated kappa agreement for the task of feature annotation in general. For this purpose, all seven lists were merged and duplicates were removed. Only the cases where items were the same and also the labels assigned by two annotators were identical were considered as duplicates and removed from the list. For example, if the item "1.3 megapixel camera" occurred in our list several times, we would check whether raters made the same judgement (e.g. both raters considered it good) and only then we would remove redundant items. However, if we had different judgements, such as the case where at first the rater considered an item to be relevant and the second time changed his/her mind, we would keep both cases. Thus 277 duplicate items were removed and general kappa was calculated for this newly compiled list. The results acquired can also be found in Table 3.4.

It was interesting to discover whether there is a correlation between how relevant the items in the lists are and how much annotators agree when annotating these lists. For this purpose, we did a similar calculation to the one presented in

No	List name	Kappa
1	**4-grams**	0.8371
2	**Bigrams**	0.7523
3	**Trigrams**	0.7362
4	**Heads of Nps**	0.6031
5	**TE-corp**	0.5507
6	**Yahoo!**	0.5317
7	**TE-doc**	0.4948
	General kappa	**0.655**

Table 3.4: Results of the inter-annotator study

Table 3.3 and counted the number of items in each list that were considered relevant by both annotators (where they agreed about the classification). The results are presented in Table 3.5.

It can be seen that the ranking of the lists due to their accuracy differs slightly from the evaluation results presented in the previous section and this is discussed in more detail in the following section.

3.2.1.5 Discussion

The evaluation confirmed our initial intuition that terminology extraction methods are more helpful than ngrams. It is worth noting that ratings (presented in Table 3.3 and Table 3.5) of the precision of methods were different. We feel that the rating which took into account only items deemed relevant by both annotators better reflect the quality of the lists. Therefore, the *TE-doc* method, which was looking for the presence or absence of a term in the document, was the best one

No	List name	Relevant items	Accuracy
1	**Trigrams**	61	20.3 %
2	**Bigrams**	69	23.0 %
3	**4-grams**	76	25.3 %
4	**Heads of Nps**	80	26.7%
5	**TE-corp**	135	45 %
6	**Yahoo!**	172	57.3%
7	**TE-doc**	**181**	**60.3 %**

Table 3.5: Accuracy of corpus-based methods for feature extraction - where two annotators agreed

for our task. This list contained a lot of terms that can be classified either as features of the phone or as their possible values. Even though the output of Yahoo! contained fewer noisy entries and even did not need additional postprocessing in order to remove some irrelevant entities. Therefore, its accuracy was higher, but a closer look at this list reveals a problem: most of the terms are names of different phones and brands. These terms are meaningful but do not meet the requirements of our research.

The previous section (Section 3.2.1.4) described the inter-annotator study and revealed that even for human annotators it is not easy to agree on what should be considered as a relevant feature/value and what should not. Value for kappa of 0.655 shows that there was a considerable number of cases where raters disagreed. An interesting observation was also made when looking at Table 3.4. The general tendency was that the more accurate the list was, more difficult it was for annotation. This observation can be used to conclude that it is easier for

the annotators to agree on which items are not relevant.

The initial analysis of the data revealed that corpus-based approach is not very helpful. Our corpus is not big enough to give good results with this methodology; for example, the trigram that had the highest frequency in our corpus was encountered only 158 times. The small amount of text did not allow us to filter irrelevant information. Nevertheless, we still found this simple corpus-based method beneficial for capturing the corpus characteristics and revealing some of the features. We believe that having a bigger corpus and using more linguistic information can help with the identification of more meaningful information. However, we would still be left with the problem of linking our features to the list of their possible values, which can be even more time-consuming than just identifying a set of features. Given that we are aiming at more automatic methods, we decided to look at this problem from another angle and see how we can diminish manual work. Another approach used for the feature identification is discussed in the next section (Section 3.2.2).

3.2.2 Semi-structured resources

After experiencing the limitations of the corpus methods described in the previous section, we considered using already available semi-structured information. Infoboxes, tabular presentations of the information about the concepts described in the Wikipedia pages, were chosen as such a kind of information. For all the products, these descriptions are made according to the features; therefore it is possible to obtain not only the features but also their corresponding values.

However, information contained in the infoboxes needs additional filtering and cleaning before it can be used for our purposes.

3.2.2.1 Features

Even though we focused on the infoboxes of the same type - "mobile phone", they were not homogenous. Initially, two types of problems with the information present in infoboxes were identified:

- Different surface forms of the same feature (e.g. "operating _ system" and "operating system");
- Functional information which cannot be considered a feature (e.g. "image").

Some of the features extracted from infoboxes were abbreviations of the full versions of the features (e.g. "os" stands for "operating system"). The identification of such situations was important for collecting a correct and consistent list of features. However, all the attempts to develop an automatic method for identifying these variations failed. For this reason, we had to perform the merging manually.

In some cases, we also found that there are spelling variations of the same features as in "operating_system" and "operating system". Several common variations were identified and mapped automatically by deleting special symbols, converting all the features to lower case, and checking for variations without spaces. This step of pre-processing helped us to compile a cleaner list of features.

We also identified that some entries in the infoboxes cannot be considered as features and are rather functional ones such as "name", "imagesize" and

"image". We are not likely to encounter these features in the texts and they are not characteristics of the phone. However, they were easily identified after examining several infoboxes, as they were positioned at the beginning of the list in the infoboxes. They were removed from the output list of the features.

The suggested heuristics helped us to remove redundant information and to merge some features in order to get a cleaner list, which was used for the further processing. These heuristics can be used for the other domains as the identified problems are common for Wikipedia infoboxes in general.

At the next stage we used the frequency of the features identified to filter out the ones that were too rare and therefore were considered as less important and reliable. Given that our thesis is focusing on employing features and values to build a resource for an IQA system, we were interested in keeping only the features that had some values openly mentioned in the infoboxes. If a feature does not have any values, it cannot be used to describe a mobile phone and therefore it cannot be employed for IQA.

Therefore the heuristics we employed can be summarised in the following way:

- check abbreviations and merge them with full forms;
- merge features by converting them to lower case, checking variations without spaces or with special characters;
- remove the functional entries of the infoboxes;
- use the frequency of the infobox entries to filter out rarely used features;
- filter the features that do not have any values.

3.2.2.2 Values

The compiled list of features also provided us with a list of values associated with them. Combination of the features and their values will be used at the later stages of our research for ranking of the features (see Section 6.3.2). Pairs of features and their values constitute the database to be used when looking for a product, in our case a mobile phone.

Values of the features have different types. Some of them indicate whether the feature exists or not and give some characteristics. For example, the feature "memory card" can have the value "yes" to indicate that a phone has a memory card or a value like "12 GB" to give precise characteristics of the memory card included. So, for example, values for the feature "rear camera" give technical characteristics of the camera: for model Nokia X6 we get a value "5.0 megapixel with Carl Zeiss optics, 4x digital zoom, dual LED flash , Front-facing CIF camera". Therefore, they can have several values even for the same phone model depending on the angle from which this feature is described. Other values can also enumerate things, e.g. the case of the feature "media", where values indicate which formats the phone is compatible with. For example, the model Nokia 5800 XpressMusic has the following value for the feature "media": "AAC, AAC+, eAAC+, MP3, MP4 (MPEG-4 Part 2 VGA / H.264 QVGA), M4A, WMA, AMR-NB, AMR-WB, Mobile XMF, SP-MIDI, MIDI Tones (poly 64), RealAudio 7,8,10, True tones, WAV, but not Ogg files". These kinds of features can introduce ambiguity, because the value of the feature for a single model of the phone is naturally not unique.

Given that we want to use the values for further processing, we are interested to get as clean and extensive lists as possible. We need to keep all the information about every phone in the database, but initially we need to split the values in order to enumerate all possible variants of the values for a given feature.

After examining closely the values of the features we found out that in several cases, not only a comma is used as a splitter of several values for the same feature, but also some other special characters, e.g.

- Special character "-": *"[[Keypad]] -[[Accelerometer]] -[[Motion sensor]] -[[LED as light sensor#Ambient light sensors]]"*;
- Special character "+": *"[[Touchscreen]] + Side buttons"*;
- Special character "/": *"[[Keypad]] / [[Touchscreen]] / [[Jog dial]] / [[Alphanumeric]]"*;
- Special character "&": *"Illuminated QWERTY [[Keypad]] & Side Thumbwheel"*;
- Special character ";": *"Type: TFT; Colors: 250K; Size: 320 x 240 pixels (2 inches)"*;
- Conjunction "and": *"[[Qwerty—QWERTY]] [[Keypad]] and [[touchscreen]]"*.

We also discovered some cases where some parts of the text describing a value were in brackets: e.g. "Bluetooth 2.0 (EDR/A2DP)". Therefore, we needed to decide whether to delete part "(EDR/A2DP)" or just consider it another value of the same feature. It was agreed that it is important to acknowledge that "Bluetooth 2.0 (EDR/A2DP)" is the same as "Bluetooth 2.0", so we need to

remove the part in brackets and keep it as a separate value for this feature.

As a result of the methods described above, a total of 229 features and 4279 values were collected; therefore, the average number of values per feature is about 18. Some of the features had the same values, so 238 values were encountered more than once. For example, the value "yes" is a very general one, and therefore it can be used for several features. The collection of features and values revealed some problems with the way infoboxes are constructed, for example, double features and values. Either simple heuristics or manual checks were used to ensure the quality of the data collected. However, these methods are general enough and can be used for processing infoboxes describing other products.

The next section (Section 3.2.2.3) will describe the comparison of the items acquired using corpus-based methods and the ones acquired from infoboxes.

3.2.2.3 Evaluation

After we extracted a list of features and values from infoboxes, it was vital to see how they coincided with the ones we acquired using corpus-based methods. The advantage of corpus-based methods is that they can be used if we have a corpus of texts describing the field, and there is no need to have semi-structured resources like Wikipedia. Therefore, it would be interesting to discover whether we can use corpus-based methods in case of a lack of infoboxes, taking into account all the shortcomings mentioned before (Section 3.2.1).

As previously mentioned, infoboxes are added by many collaborators and therefore are a reliable source of information. We can consider the collected list of

features and values as a gold standard and calculate the recall to see how many items from infoboxes can be found using corpus-based methods.

For this purpose the list acquired from infoboxes was additionally cleaned: we removed duplicate values from the list. As a result of this, we obtained a list of a total of 4270 items. As for the list extracted using corpus-based methods, we have taken only the items that both annotators considered relevant. It gave us a list of 565 items in total.

At the next step we examined these two lists to find cases where there was overlap between them. We investigated two cases: when we had an exact match in two lists (e.g. we saw item "email" or "xenon flash" in both lists) and when lists contained similar items (e.g. "2.0 mp" would be considered similar to "2.0 mpx").

206 items from the corpus-based list had identical items in the infobox list and 117 items were considered similar. So in total, 323 items out of 565 were present in the infobox list. This gives us 57.2% recall with respect to the corpus-based list.

We examined items that were not found in infoboxes to see whether there were some regularities to explain why some items were not present in the infobox list. We managed to identify that there are many cases where the list collected using corpus-based methods contained more general items than infoboxes that featured more concrete values. For example, the corpus-based list had "microsoft windows mobile" whilst the infobox list contained more specific items such as: "microsoft windows mobile 6.0 standard", "microsoft windows mobile 6.1 professional", "microsoft windows mobile 6.5 standard". However, there were situations where

the infobox list contained more general items and the corpus-based list more specific ones, such as "camera features" in the infobox list and "camera interface", "camera lens", "camera sensor" in the corpus-based list. Therefore, we can conclude that the recall can be considered higher if we regard these cases as similar items.

3.3 Conclusions

In this chapter we presented the data to be used in this thesis, specifically two types of corpora: semi-structured texts (Section 3.1.1) and unstructured texts (Section 3.1.2). This chapter provided more details about the way these corpora were collected. Both corpora are used for the development of coreference resolution methods described in Chapter 4. The second corpus is also used for the identification of the links between mentions of the phone in the text and features, which is described in more detail in Chapter 5. This chapter also focused on automatic acquisition of phone features using corpus-based methods (Section 3.2.1) and semi-structured resources (Section 3.2.2). The next Chapter describes the development of the coreference resolution methods and their evaluation.

We managed to clean the features and corresponding values we had extracted from infoboxes and evaluated whether corpus-based methods can be used in case of lack of infoboxes. For the research carried out in this thesis, it was decided to employ lists collected using infoboxes as they allow us to have product features and also values connected to them.

Chapter 4

Coreference

4.1 Introduction

The main aim of information extraction is to extract as much reliable information as possible. However, it can be problematic if the computer does not know which information is relevant to the task in question. For example, if we need to extract some information about a mobile phone, it would be obvious for a human being which information is about this entity and which refers to another phone. However, automatic systems could find it difficult to distinguish between them, so the step of coreference resolution, organising entities in coreferential chains, becomes crucial. More details about the notions of information extraction and coreference resolution are provided in Section 4.2.1.

An extract from our corpus describing a mobile phone, model name "Nexus One", exemplifies the problems automatic methods can encounter when analysing the text:

> *[The Nexus One]*$_1$ (codenamed *['HTC Passion']*$_2$) is *[Google's flagship smartphone]*$_3$ manufactured by Taiwan's HTC Corporation. *[It]*$_4$ became available on January 5, 2010 and uses the Android open source

mobile operating system. Features of *[the phone]*$_5$ include the ability to transcribe voice to text, noise canceling dual microphones, and GPS guided turn-by-turn voice directions to drivers.

[The device]$_6$ is sold unlocked (not restricted to use on a single network provider). Google currently offers *[it]*$_7$ for use on the T-Mobile and AT&T networks in the United States; *[a version for use on Vodafone (European) networks]*$_8$ was announced on 26 April 2010, available in the UK on 30 April 2010.

This text reveals that coreference resolution is very important for the task addressed in this research. In order to extract information about "Nexus One" the system should capture when new information connected to it appears in the text. However, it can be difficult given that "Nexus One" is referred to throughout the text in several ways: "it", "the phone", "the device". However, NP$_8$, although in close proximity to other NPs describing "Nexus One", is not really part of the chain. Therefore the quality of information extraction will decrease if these facts are not taken into account. Thus the problem of coreference resolution should be addressed in order to improve the quality of the information extraction.

This chapter will address the problem of coreference resolution for two types of corpora described in Section 3.1. Firstly, we will discuss in more detail the field of information extraction and coreference resolution (Section 4.2.1). Subsequently, we will explain the motivation to develop our own coreference resolution system rather than use already existing systems (Section 4.3). Section 4.4 will describe

the annotation guidelines and the process of corpus annotation. Section 4.5 will address the development of the coreference resolution algorithm for Wikipedia texts (Section 4.5.1) and the texts from the review domain (Section 4.5.2). The system's evaluation is described in Section 4.6. It is followed by Section 4.7, which details the error analysis of the algorithm suggested for coreference resolution. The chapter ends with the conclusions presented in Section 4.8.

4.2 Related Research

4.2.1 Information extraction

Information extraction (IE) is a field of computational linguistics which plays a crucial role in the efficient management of data. It is defined as "a process of getting structured data from unstructured information in the text" (Jurafsky and Martin, 2009). Grishman (1997) describes this process as "the identification of instances of a particular class of events or relationships in a natural language text, and the extraction of the relevant arguments of the event or relationship". After the information is structured and added to a database it can be used by a wide range of NLP applications, including information retrieval, question answering and many others.

Information extraction challenge has a long history and goes back to the late 1970s (Cowie and Lehnert, 1996); however the first commercial systems appeared only in the 1990s, e.g. JASPER (Andersen et al., 1992), specially built for Reuters. Later research was greatly inspired by a series of Message Understanding

Conferences (MUC)[1], which were initiated and financed by the Defense Advanced Research Projects Agency (DARPA) to encourage the development of new methods in information extraction. The importance of the MUCs was not the conferences themselves, but the evaluations and evaluation competitions they proposed (Grishman and Sundheim, 1996). The organisers of these conferences defined tasks for all the participants, prepared the data and developed the evaluation framework for each task. Researchers had to address the task and find the best solution; therefore it added competition element to the research. In addition to all the above-mentioned advantages, these events were an opportunity to get comparable results and evaluate objectively the performance of different systems. MUCs were followed by several ACE (Automatic Content Extraction)[2] evaluations which also provided valuable feedback for researchers.

Usually IE can be regarded as a pipeline process, where some kind of information is extracted at each stage. Jurafsky and Martin (2009) point out several different types of information that can be extracted:

- named entities (NE);
- temporal expressions;
- numeric values;
- relations between entities and expressions previously identified;
- events/template filling.

Generally IE starts with the detection and classification of proper names found

[1]http://www.itl.nist.gov/iaui/894.02/related_projects/muc/
[2]http://www.itl.nist.gov/iad/894.01/tests/ace/

in the text, which is usually referred to as Named Entity Recognition (NER). Most commonly IE systems search for names of people, companies and organisations, and geographical places. But the choice of the precise kind of NE to be extracted depends greatly on the task and system in mind. Sometimes the notion of Named Entities is extended to include items that are not really names or entities, but bear important information for analysing the texts; therefore, numeric values, such as measurements and prices, or temporal expressions can be included in this category (Jurafsky and Martin, 2009). Extraction of such kinds of data is extremely important for correct analysis of texts and reasoning.

Usually the next step in IE is coreference resolution, the identification of identity relations between Named Entities (Jurafsky and Martin, 2009). At this stage, mentions of the same Named Entity, which are expressed using different linguistic realisations, are found. The process of coreference resolution is crucial for getting more accurate results in IE and more details about this process are provided in the next section.

Relation extraction is a step further in analysing information in the texts and turning unstructured information into structured information. This stage involves identifying the links between Named Entities and deciding which ones are meaningful for the concrete application or problem. Relation extraction is an important part of this research and is discussed in more detail in Chapter 5, which deals with identification of relations between mentions of the phone and its features.

The next stage of information extraction is template filling. Template filling

involves extracting appropriate material to fill in the slots in templates for some stereotypical situations that recur quite often. For example, we can be interested in extracting information about some terrorist attack and this event can be treated as a template, which has predefined slots: place, date, number of people injured/killed, organisation who took responsibility for the terrorist act, etc.

Given the approach used in this research only the work in coreference and relation extraction is directly relevant.

4.2.2 Coreference resolution

As mentioned earlier coreference resolution can be regarded as a step in the information extraction pipeline and is important in enhancing the performance of the subsequent steps. This task had already aroused the interest of researchers in the 1960s-70s, but in the 1990s it experienced a revival, when the NLP community moved from using heuristical methods to machine learning. Even though hand-crafted methods were useful for getting quick results and were easy and transparent for further adaption, they turned out to be domain-dependent and at times difficult to apply to general texts (Vieira and Poesio, 2000; Muñoz et al., 2002). Additionally, the development of the coreference resolution field received a boost as a result of several conferences, such as MUC-6 (1995) and MUC-7 (1998), that provided annotated corpora and the possibility of using common evaluation schemes. Specialised conferences such as Discourse Anaphora and Anaphor Resolution Colloquium (DAARC) and various workshops like SemEval[3]

[3]http://www.senseval.org/

have also contributed to the advance of coreference resolution research.

Nowadays there are a lot of systems that deal with coreference resolution. As mentioned above, the most recent systems are based on machine learning techniques. Ng (2010) enumerates three kinds of learning-based coreference models: the mention-pair model, the entity-mention model and ranking models.

4.2.2.1 Mention-pair model

The mention-pair model is the most common model and Recasens and Hovy (2009) point out that it splits the problem of coreference resolution into two steps when using learning-based approaches:

1. classification, in which a classifier is trained on a corpus to learn the probability that a pair of NPs are coreferent or not;
2. clustering, in which the pairwise links identified at the first stage are merged to form distinct coreference chains.

The first step is based on machine learning and the number of features used can vary a lot: from 12 (Soon et al., 2001) to 351 (Uryupina, 2007). Usually all the features used at this stage can be classified in several classes:

- lexical (e.g. match of synsets in WordNet; whether entities are pronouns or not);
- syntactic (e.g. syntactic function of the entities);
- discourse (e.g. whether mentions are part of direct speech).

Recasens and Hovy (2009) also add language-specific and corpus features. Ng (2010) suggests more fine-grained classification and categorises all features into those that are string-matching, syntactic, grammatical, semantic, discourse-based and all other features.

The top feature for coreference is head match and is quite often used as a baseline, but Elsner and Charniak (2010) point out that NPs with the same head are not necessarily coreferential and making the assumption that all same-head pairs corefer can introduce a considerable amount of errors. This was observed while trying to develop a topic tracking method for Wikipedia articles which is described later in this chapter.

Other features commonly used in coreference resolution include kinds of semantic similarity (e.g. distance in WordNet), types of NPs involved in the relation (e.g. common nouns, pronouns, NEs), types of NE, and the gender and number agreements of the mentions. The set of features and their number can vary a lot depending on the domain and the type of texts processed.

After the first step of the coreference resolution is done, all pairwise classification decisions should be used for NP partition and organising NPs into coreference chains. The most commonly used coreference clustering algorithms were proposed by Soon et al. (2001)(closest-first clustering) and Ng and Cardie (2002b)(best-first clustering). However the major problem of these methods is their greediness: they tend to merge into clusters of too many instances and favour positive pairwise relations over negative ones. There have been several attempts to overcome these problems by using other methods: correlation clustering

(Bansal et al., 2002; Finley and Joachims, 2005) and graph partitioning algorithms (McCallum and Wellner, 2004). It is difficult to say which algorithm yields the best results, because no proper comparison was done between them.

Determining the anaphoricity of an NP is another problem that has to be solved independently in order to improve coreference resolution. Research shows that training a classifier to filter non-anaphoric NPs prior to coreference resolution can improve the results of the coreference resolution (Poesio et al., 2004; Ng and Cardie, 2002a).

4.2.2.2 Entity-mention and ranking models

Two other models mentioned above, the entity-mention model and ranking model, are less popular and are used only by a small fraction of systems. The entity-mention model tries to "learn a model that can classify whether an NP to be resolved is coreferent with a preceding, possibly partially-formed, cluster" (Ng, 2010). This model has not been exploited widely enough yet (Ng, 2008; Culotta et al., 2007). Its theoretical description promises better results; however its advantage over the mention-pair model was not proved in real implementations.

Ranking models provide information about which candidate is the most probable antecedent for a chosen NP. This method allows ranking of all of the candidate antecedents simultaneously. Even though it showed better results than the mention-pair model (Denis and Baldridge, 2007), this model usually does not use cluster-level features and so loses valuable information. Rahman and Ng (2009) tried to address this problem by proposing a cluster-ranking model, which ranks

preceding clusters, rather than merely candidate antecedents. This approach seems promising; however, not much research has been done in this direction so far.

4.2.2.3 Systems

There are currently several systems offering coreference resolution for all kinds of texts, not restricted to some specific domains: RECONCILE (Stoyanov et al., 2010), BART (Versley et al., 2008), Stanford Deterministic Coreference Resolution System (Raghunathan et al., 2010). They are based on machine-learning approaches and are trained based on corpora annotated with coreferential relations. Usually these texts belong to the newspaper domain and therefore even though systems can work for any domain "out of the box", giving opportunity to get results quickly, the output is not always of high quality without having been adapted. In order to get better quality results for a specific domain, these systems have to be adapted and retrained for more specific texts. This can be difficult to achieve when there are no big domain specific corpora annotated with coreferential relations. Therefore it is a matter of weighting up the pros and cons when deciding whether to use such a system for special domains or whether to develop your own.

State-of-the art systems report an MUC score of about 70-80% (Recasens and Hovy, 2009), (Stoyanov et al., 2010), so there is still much space for improvement and enhancing modern coreference resolution systems. However, Ng (2010) mentions that it is difficult to estimate how well systems perform and compare because most of them use different evaluation metrics and different data sets, which makes objective comparison a difficult task.

4.3 Motivation

As discussed in Chapter 1, one of the hypotheses of this research is that sentences which are not directly related to the phone and its features can introduce noise in ranking. Therefore, we are interested in finding all sentences that contain mention of the phone as well as a phone feature. In order to achieve this, as many mentions of the phone in the given text as possible should be identified. Coreference resolution can link mentions of the phone in the text in one chain, thus ensuring that we choose only the relevant sentences for further processing.

As mentioned in Chapter 3, we deal with two types of texts and therefore we need to develop coreference resolution methods that would be suitable for both types.

Initial examination of the texts showed that they have a number of special characteristics. For example, one particularity of the Wikipedia articles is that they focus on only one topic (e.g. a product, person, location or event), which is detailed throughout the article. Therefore in order to extract comprehensive information from these articles, it is necessary to be able to track different expressions that refer to the topic. Once all different ways of referring to the same topic are identified, we can proceed to relation extraction. This explains why we are not interested in resolving coreference for all words in the text and only pay attention to the chains containing the main topic of the article.

Attempts to use state-of-the art systems for coreference resolution showed that they provide very low precision for the task in question and link mentions which

are not coreferential at all. In most cases it happens because the algorithms rely heavily on substring matching and distinguish rather poorly between entities with similar names. It can be clearly observed when examining the chain generated by RECONCILE (Stoyanov et al., 2010) for the article describing the mobile phone "HTC Magic": *"The HTC Magic" - "HTC" - "The HTC Dream" - "Vodafone" - "it" - "the Vodafone Magic"*. The state-of-the-art systems focus on identifying all chains in the text and therefore there are high chances that the chain referring to the topic, which in our case is the phone, will be broken into several chains. Futhermore, as described in Section 4.4.2, we adapt a slightly different notion of coreference, which introduces additional difficulty in using state-of-the-art systems. The above-mentioned reasons provided us with a motivation for developing our own system that will work with high accuracy for our specific domain. The developed system and its evaluation are described in the following sections.

4.4 Corpus annotation

As previously discussed, in order to be able to employ NLP techniques and explore ways of getting information semi-automatically, we need to annotate the corpus that will be used at later stages for information processing and for evaluation of the methods developed.

Section 3.1 already described two corpora that were collected for this research: Wikipedia texts and user reviews. However, it should be mentioned that initially our research involved only the use of Wikipedia articles, and it was not until at the later stages that we discovered that we needed more data for our experiments.

Therefore, our investigation started with the Wikipedia corpus and the review corpus was acquired later.

The next section describes the annotation tool that was use to annotate our corpora (Section 4.4.1). Subsequently, the relations that were introduced and later annotated in our corpora are discussed in Section 4.4.2.1. Sections 4.4.2.2 and 4.4.2.3 describe the annotation of Wikipedia texts and the user review texts accordingly.

4.4.1 Annotation tool

To speed up the annotation and ensure its consistency the annotation tool PALinkA (Orăsan, 2003) was used. It is a language- and task-independent tool which allows you to define your own link types. Users can benefit from its intuitive graphical interface which does not require complicated training and is easy to use. The output of this program is a valid xml document, which makes the following processing easier. The users do not need any technical education; the tool itself prevents them from introducing mistakes into the xml file structure.

The tool allowed us to annotate markables in the text and at the next step we were able to add relations to indicate links between several markables. Although the graphical interface does not show xml tags in the texts, it uses colours to denote the markables and relations, which makes annotation representative and easy to analyse and correct if needed. Therefore the use of this annotation tool made the annotation process easier and more transparent.

4.4.2 Coreference annotation

As previously described we had two types of texts to be annotated with coreference: Wikipedia and the review articles. Initially our investigation started with the Wikipedia corpus. It was the Wikipedia corpus that was used to develop the guidelines and to identify the relations of interest. After we discovered that we needed more data and different sources of information, the existing guidelines were adapted to take into account all peculiarities of the review domain.

The next section focuses on the description of the relations to be identified and annotated in terms of the coreference task.

4.4.2.1 The notions of coreference and near identity

Noun phrase (NP) coreference resolution is usually defined as "the task of determining which NPs in a text or dialogue refer to the same real-world entity"(Ng, 2010). Coreference resolution overlaps with the field of anaphora resolution, but there is one main difference between them: anaphora is "pointing back to a previously mentioned item in the text" and coreference is "the act of referring to the same referent in the real world" (Mitkov, 2002).

In view of the above definitions, it is obvious that some links between words in the text can be anaphoric, but not coreferential. Hirst (1981) distinguishes between *Identity of Reference Anaphora*, where anaphor denotes the same entity as its antecedent, and *Identity of Sense Anaphora* which denotes a relation not between the same entities, but ones of a similar description. The former corresponds to a coreferential relation, whilst the latter does not. Anaphoric links can also be

classified as *direct* and *indirect* (also known as bridging or associative) anaphora (Mitkov, 2002). Direct anaphora links anaphors and antecedents by relations such as identity, synonymy, specialisation or generalisation. In contrast, indirect anaphora links anaphors and antecedents by relations such as meronymy/holonymy or set/subset[4].

The classical definition of coreference presupposes that entities can be either coreferential or not. However, recent research (Recasens et al., 2010) shows that this definition covers only a very specific type of relation and a much more fine-grained definition should be used instead. We encountered the same problem while investigating a corpus of Wikipedia pages with the purpose of annotating coreference relations. One of the features of Wikipedia articles is that they have a unique topic throughout the whole article, for example, the article about "BMW E46" should focus on this model of the car. However, corpus investigation showed that it is not always easy to track this topic by simply relying on the identity relation.

To address the above problem, a corpus of Wikipedia articles was analysed. On the basis of this corpus investigation, we decided to focus on 4 types of relations that are useful for our IQA task.

- **Coreference**: This corresponds to the classical notion of coreference as defined by (Ng, 2010). This relation is the most frequent one and forms transitive coreference chains. Simple coreference should be carefully

[4]Here Mitkov's terminology is used, but (Poesio and Vieira, 1998) classifies anaphors in a different way, separating coreferential relations and bridging ones

distinguished from relations SET OF and SIBLINGS (presented below), as sometimes the distinction between them is not straightforward.

Example:

> *[The HTC Evo 4G]* (trademarked in capitals as *[EVO 4G]*) is *[a smartphone]* developed by the HTC Corporation that was previously known by *[its]* codename *["Supersonic"]*. *[The smartphone]* launched June 4th, 2010. *[It]* is *[the first 4G capable phone]* sold in the United States. *[It]* became the top-selling launch day phone on Sprint, surpassing the Palm Pre.

- **Set of**: One characteristic of the Wikipedia pages discussing products is that they can describe several versions of the same product. This is normally marked by adding a prefix or suffix to the original name. Given the purpose of this research, such links should be identified in texts, but they should not be marked as identity as they refer to entities with different characteristics. For this reason, we add a SET OF link from the markable to the antecedent that describes the set (i.e. the topic of the article).

  Example:

  > A modified version of *[the Hero]*, *[the HTC Droid Eris]*, was released on the Verizon Wireless network on November 6, 2009.

- **Alias**: Another characteristic of Wikipedia product articles is that the same product can be referred to using different names. This is a special case of coreference relation where a completely different name is used for the product

and not a substring of the original name. This relation is usually indicated by phrases such as *is also named as* and *has codename*.

Example:

> *[The HTC Touch Diamond]*, also known as *[the HTC P3700]* or *[its]* codename *[the HTC Diamond]*, is a Windows Mobile 6.1-powered Pocket PC designed and manufactured by HTC.

- **Siblings**: For interactive question answering it is very important to identify when two entities differ in terms of only a few characteristics. This is due to the fact that in the case of ambiguity a user should be presented with close alternatives and be asked to decide between them. This relation normally happens when the two entities are in a SET OF relations with the topic of the article. We call the link between these entities SIBLINGS relation to indicate the near-identity between them. In our corpus this phenomenon happens quite often when the same mobile phone is distributed by different operators with slightly different features, and possibly with a different name.

  Example:

  > *[The BlackBerry Bold 9700]* (codenamed *["Onyx"]*) is *[a high-end mobile phone data device]* (smartphone) developed by Research In Motion.
  >
  > *[The BlackBerry Bold 9650]* is *[the newest device]* in the BlackBerry Bold series.

The next section presents how these relations were annotated in our corpus.

4.4.2.2 Annotation of Wikipedia texts

For this research we used part of the Wikipedia corpus that was described earlier (Section 3.1.1). Articles from the domain of products (and more specifically about mobile phones) were annotated with the relations described in the previous section.

Currently the annotated corpus consists of 20 documents with almost 22,000 words. To enable the annotation process, clear guidelines were developed to maximise the inter-annotator agreement. Since traditional guidelines do not cover all the situations we encountered in our domain, we had to adapt the existing guidelines (Hasler et al., 2006) and change the notion of coreference. The remainder of this section presents the annotation guidelines used to mark the relations presented above.

As proposed in (Hasler et al., 2006), the first step of the annotation process was to mark all the NPs, including the embedded NPs, pronouns, definite descriptions and proper names, as mentions (e.g. *it, the device, The HTC Touch Diamond*). This was done regardless of whether they were linked to the topic or not, and was achieved using PAlinkA, mentioned in Section 4.4.1. Our corpus contains a total of 3372 markables. The second step was to mark links between these markables as described below.

The coreference relation

The COREFERENCE relation is marked only between markables that refer to the same entity in the real world. This includes coreferential links such as identity, synonymy, generalisation and specialisation, but they were not explicitly distinguished as proposed in (Hasler et al., 2006). In general, only definite descriptions that stand in the relationship of identity (same head: *a smart phone - The Touch Pro smartphone*; pronouns: *Opera Mobile - it*) or synonymy (*the device - the phone*) with the antecedent were marked as coreferential. Usually an anaphoric expression is linked to the previous mention of the NP in the document, but it can be also linked to the first mention as the relation is transitive.

Text in brackets and text between dashes after an NP is marked as coreferential with this NP (as long as it definitely refers to the NP): e.g. *[the XV6800 ([Verizon Wireless]) variant of [the device]]*. For this type of coreferential link, the anaphor should be linked back to the nearest antecedent in the document.

Set of

The SET OF relation is used to link members of hyperonymy hierarchy: it links a less general markable to a more general one. For our corpus, this happens when a phone has several submodels. The link is always added from the submodel to the nearest markable that corefers to the topic. SET OF is also used to identify more general categories than the topic as it happens to markables in copular

relation like in the following example: *[The HTC Dream] is [an Internet-enabled 3G smartphone].* In this case the relation will be from *"The HTC Dream"* to *"an Internet -enabled 3G smartphone"*. This makes it possible to collect more information about the topic.

Alias

Relation ALIAS is quite straightforward and is used to indicate situations when different names are used for the same entity. This relation is quite common in our corpus and usually is introduced by a limited set of verbal phrases. The link is always from the markable that represents the alias to the nearest markable that corefers with the topic.

Siblings

This relation is not explicitly marked during the annotation process, but it can be inferred on the basis of the above annotation.

In our corpus we annotated a total of 668 coreferential relations, 83 SET OF relations and 59 ALIAS relations.

Example of annotation

The example below shows an extract from the text describing the HTC Magic and various types of relations between entities. In order to keep the text legible not all the markables and relations between them are highlighted.

> *[The HTC Magic]*$_{1,1-set-of-4}$ (known as *[the T-Mobile myTouch 3G]*$_{2,2-alias-1}$ in the US, and *[the docomo HT-03A]*$_{3,3-alias-1}$ in Japan) is *[a smartphone]*$_4$ designed by HTC.
>
> [...]
>
> *[The T-Mobile myTouch 3G]*$_{5,5-set-of-1}$ comes bundled with a pair of headphones, an extUSB headphone adapter (which also serves as an in-line microphone), wall charger, USB Cable, Cloth Pouch, Screen Protector, and documentation, all inside a unique carrying case that also serves as the retail box. *[The Phone]*$_{6,6-coref-5}$ itself also has a 4GB SanDisk Class 2 microSD Card (SDHC) inside.
>
> *[The Vodafone HTC Magic]*$_{7,7-set-of-1}$ comes bundled with a pair of headphones, wall charger, USB cable, leatherette pouch and documentation. *[The phone]*$_{8,\ 8-coref-7}$ comes with a 2GB microSD card.
>
> *[The Vodafone Germany/New Zealand/Australia HTC Magic]*$_{9,9-set-of-1}$ comes bundled with a pair of headphones, wall charger, USB cable, leatherette pouch and documentation. *[The*

$phone]_{10,\ 10-coref-9}$ comes with a 8GB microSD card.

The first sentence of the extract introduces "The HTC Magic" phone and it is the topic.

It can be observed that every provider sells phones with different features and it should be taken into account in terms of IQA. However, only the phone that is linked to T-Mobile gets a new name: "myTouch 3G", and the other two still have the name "HTC Magic", but have different characteristics. Therefore, the two notions will be marked as "sibling concepts" because they have the general parent "HTC Magic" which is not mentioned here and are just two children of this general concept.

The Wikipedia corpus annotated with coreference relations was further used for developing a coreference resolution method that is described in Section 4.5.

4.4.2.3 Annotation of review texts

As previously mentioned, we initially annotated the Wikipedia corpus and only at the later stages of our research we did we realise the need for additional data. Therefore it seemed logical to adapt the guidelines already developed for Wikipedia and follow the same steps when annotating the reviews. This strategy should also guarantee that the results obtained using two different corpora will be comparable.

The annotated corpus consists of 20 documents with almost 77,500 words. We aimed to annotate similar amount of files to the Wikipedia corpus in order to make the results comparable. However, it can be seen that the review texts were longer. In total 1125 markables were annotated.

Wikipedia and the review corpora share quite a lot of similarities, and therefore, it was decided to use the same annotation guidelines for both of them. Annotation was done in a similar manner; however, for the review corpus, not all markables were annotated, only the ones describing the mobile phone in question. For the annotation of the review corpus, the same relations were used as were used in annotating the Wikipedia corpus. However, during the annotation, we found that the coreference relation was the most widespread, and that other relations were not very common in the review corpus. The statistics acquired from the annotated corpus revealed that we annotated 1075 coreferential relations, 19 "set of" relations and only two "alias" relations.

It should be mentioned that in the case of the review corpus "set of" relations were used in most cases to indicate that the phone belongs to a more general group of objects. For example, we can argue that in the sentence "This phone is a Samsung smartphone" "this phone" is linked to "Samsung smartphone" by "set of" relation. This phone is one of many Samsung phones.

Annotation of the review domain was in some sense easier than that of Wikipedia articles. This can be explained by the fact that the majority of the relations were coreferential and were easier to identify. For example, the review corpus did not have the description of a set of products like in Wikipedia. In most cases the whole review focused on one phone someone owned and other products were not frequently mentioned; therefore, it was easier to identify the coreferential chain and the annotation was more transparent.

However, several decisions had to be made in order to get a consistent

annotation. For example, we had to decide how to treat cases where the author is discussing the model of the phone in general and then switches to talking about the phone he/she owned. It was concluded that we would regard these items as coreferential, because we are mostly interested in finding characteristics of the phone model discussed in the review and these items should share the same characteristics. Therefore we would ignore the fact that these are different items in the real world. The same decision was taken for the case involving refurbished phones: we considered the new phone and the refurbished one as coreferential.

It became clear that the reviews were less technical than the Wikipedia corpus and have less factual information. The review texts are lexically richer, and therefore the phone can be referred to in a greater variety of ways. For example, in addition to markables usually used in the Wikipedia corpus like "the phone", "the device", we can also encounter less formal variants like "your phone", "my phone". We anticipate that it can result in coreference resolution being less effective for the review corpus because it would be more difficult to list all the possible ways the phone can be referred to in the text.

The corpora annotated for coreference were used for the development and evaluation of the coreference resolution algorithm that is described in more detail in the next section.

4.5 A rule-based coreference resolution method

The corpus annotation described in the previous section revealed some regularities in the way expressions refer to the topic which could be captured using a rule-

based approach. This section will briefly present these rules followed by the evaluation results (Section 4.6). As previously mentioned, the initial work focused on Wikipedia texts; however at the later stages of this research we added the review texts as well. This is reflected in the structure of this section: Section 4.5.1 describes the algorithm developed for Wikipedia texts and the next section (Section 4.5.2) outlines the adaptation that was done in order to use the suggested method for the review texts.

4.5.1 Wikipedia texts

Examination of the Wikipedia texts describing products revealed several peculiarities, that make it possible to develop a rule-based method that relies on high precision rules. Various rules are used to target the different types of coreference relations described in Section 4.4.2. Given that our current focus is on the identification of expressions that refer or are linked to the main topic of the article, we initially relied on the markables annotated by humans. This allowed us to ensure that no errors were introduced in the process as a result of wrongly identified markables.

The identification of all the relations was combined into a pipe-line, where already identified relations were used for further processing. First of all, ALIAS relations were found in the text and alternative names of the topic were added to the list. This helped to reveal all possible ways the topic can be referred to throughout the text. Given the fact that we were interested not only in tracking the topic but also all subtopics, the next step was the identification of SET OF

relations. This stage yielded a list of subtopics and at a later stage they were treated in a similar way to topic expressions in order to identify all coreference chains related to the topic. The last step was focused on discovering all coreference links for topic and subtopics.

The following list shows a few examples of rules used to identify the relations of interest:

- Expressions such as *also called, formerly known as* between two markables indicate that the second markable is an alias for the first one;
- If the topic is included in a longer markable, the relation between the markable and the topic is SET OF e.g. the markable *The GSM BlackBerry Storm* is in the relation of SET OF with the topic *The BlackBerry Storm*;
- A markable corefers with the topic if the topic ends with the markable after the determiners are removed e.g. the markable *the Bold 9700* corefers with the topic *The Blackberry Bold 9700*;
- A number of NPs (*the phone*, *the device* etc.) are linked to the nearest preceding markable, if this markable refers to the topic or subtopic and is found in the window of five sentences. These noun phrases were collected by observing regularities in the text during the annotation phase;
- Pronoun *it* is linked to the nearest preceding markable, if this markable refers to the topic or subtopic and is found in the window of two sentences.

These rules were developed on the basis of corpus investigation that revealed some regularities in the ways that names of submodels are formed and names of the

phone are abbreviated. Usually the beginning of the long name denoting a phone can be omitted: e.g. *"The Nokia N900"* and *"The N900"*, *"Motorola FONE F3"* and *"the F3"*, *"The Motorola Droid"* and *"The Droid"*. Names of the submodels are in most cases a combination of the name of a general model and some number or noun phrase, e.g. *"iPhone"* and *"iPhone 3G"*. It was discovered that using simple substring matching can introduce a considerable amount of errors: it links incorrectly in one coreference chain the phone and its submodels, as well as the phone and its manufacturer. Therefore, to find all mentions of the phone in the text, we should match only the last part of the expression.

The use of these rules allowed us to identify chains referring to the topic of the article, the phone. The results of the evaluation of the method are presented in Section 4.6.

4.5.2 Adaptation to the review domain

As mentioned above, we initially dealt with Wikipedia texts describing mobile phones and only at the later stages included the review texts. Therefore, we wanted to test whether it was possible to adapt the already developed method to be used for other kind of texts. If we are able to prove that such an adaptation is feasible and does not require much effort, we can assume that it is possible to use the suggested method for other kinds of texts after a slight adaptation.

The annotation of markables was the first issue we had to address. As described in the previous section, we relied on manual annotation of markables for Wikipedia texts. Using manual annotation can be time-consuming and we therefore had to

seek automatic ways of annotating the markables. For this purpose we annotated texts with the help of Machinese (Tapanainen and Järvinen, 1997) and then we enriched the annotation and explicitly marked the boundaries of the noun phrases (NPs). We considered all NPs, including embedded ones, as markables. This allowed us to move away from manual annotation and employ the automatic methods instead. However, we acknowledge that it introduced additional mistakes. This problem is discussed in more detail in Section 4.7 dedicated to error analysis.

We discovered that the review texts are less formal, and, therefore, there are fewer regularities in the way phones are referred to. For this reason, we had to use world knowledge in our algorithm and add names of the phones in a special dictionary. Compilation of such a dictionary can be done semi-automatically: the title of the review file contains the name of the model. However, at the next stage, other expressions that were most frequently used to refer to this model were found by querying the web with the name of the phone.

After examining the review texts, we noticed that the SET and ALIAS relations were quite rare, and the annotation statistics provided in Section 4.4.2.3 confirms this observation: in the annotated corpus of 77,500 words, only 19 "set of" and two "alias" relations were encountered. As mentioned in the description of the corpora (Section 4.4.2.3), the review texts tend to discuss one model of the phone that the user is reviewing. Therefore, due to the nature of the review domain, there are fewer cases where submodels are discussed in the review. Thus we did not find the SET relation useful for the review domain.

In the texts belonging to the review domain, the users are more likely to focus

on their experiences with the product, rather than go into a lot of technical details, which would also usually include listing all alternative names the phone can be referred to. Also, for the identification of the ALIAS relations, we needed to have already marked the first mention of the topic in order to find other ways the phone can be referred to. Wikipedia's predefined structure would tell us exactly the first mention of the phone; however, in the review files, we lacked this information. Therefore, it was difficult to use heuristics to identify this relation. However, as mentioned before, for the review domain we use world knowledge and therefore identify all synonyms used to refer to a phone.

Hence, the algorithm focused solely on identification of the REF relations and tracking all the mentions of the topic in the text.

The language of the review domain tends to be less formal and more personal than the one used in Wikipedia texts. Therefore, we had to modify our algorithm to account for these differences and enlarge the list of definite NPs that can refer to the phone reviewed. Such NPs regularly referring to the phone reviewed include "my phone", "my device", "your handset" etc.

We tried using different distance thresholds when working with definite NPs to identify a possible link to the topic chain. In the case of Wikipedia articles threshold distance is needed to account for the fact that the article can discuss series of phones and provide information about similar models. However, the experiments revealed that for the review domain there is no need to impose any threshold. The addition of the threshold just causes the drop of the recall, which can be explained by the fact that in some reviews there are large gaps between

the mentions of the phone, and using the distance threshold results in ignoring candidates that are situated too far away.

The algorithm developed for Wikipedia articles had also introduced the distance threshold when dealing with pronoun 'it', which was 2. We tried experimenting with different thresholds for the review texts and discovered that the threshold 1 (i.e. the nearest preceding markable belonging to the topic chain should not be more than one sentence away from the pronoun 'it') yields the best results.

4.6 Evaluation

The evaluation process was carried out in a similar way for both types of texts, but some adaptation was made for the review domain. We attempted to make the evaluation settings as similar as possible in order to be able to compare the results.

We used the MUC score (Vilain et al., 1995) to assess the accuracy of the topic identification for both kinds of text. This score is based on comparing equivalence classes defined by the links in the gold standard and in the output of the system. The recall (respectively precision) is calculated by identifying the least number of links that need to be added to the system's output (respectively the gold standard) so that the classes will be aligned. MUC is sometimes criticised because it does not "punish" the system for identifying too many single classes (Recasens and Hovy, 2011); however, due to the nature of our research, it was not considered a problem. Therefore, we decided to use the most classical metric, MUC, which is

better understood and more widely used.

4.6.1 Wikipedia texts

We used the MUC score to evaluate the performance of the rule-based system developed for coreference resolution for Wikipedia. However, following this method, the chains that refer to subtopics of the article, e.g. submodels of the phone discussed, were not taken into account. Therefore it was decided to evaluate SET OF as well as ALIAS relations separately. It was done in terms of accuracy: how many of the identified relations were identified correctly. Accuracy of SET OF relation achieved only 11.1 %, but this can be explained by the difficult nature of such relations. ALIAS relations were identified with an accuracy of 57.9 %.

As mentioned above, the main assumption of our research is that Wikipedia articles describe a topic and provide more information about it. Therefore as a baseline all subjects of the sentence in the corpus were annotated as coreferential with the topic. Machinese[5] (Tapanainen and Järvinen, 1997) was employed for annotation of the corpus with syntactic relations and then tag SUBJ was used to identify all sentence subjects in the text.

Evaluation of the system output showed that in the case of dealing only with definite noun phrases, our system can achieve 88.24% f-measure, whereas the baseline gets only 34.34% f-measure (Table 4.1). However, when we also try to identify pronouns that are coreferential with the topic, f-measure drops to 87.15%. This issue is discussed in more detail in Section 4.7 which deals with error analysis.

[5]http://www.connexor.eu/technology/machinese/machinesesyntax/index.html

	Rule-based: only NPs			Rule-based: with pronouns			Baseline		
File	**Precision**	**Recall**	**F-measure**	**Precision**	**Recall**	**F-measure**	**Precision**	**Recall**	**F-measure**
BlackBerry Bold	80.95	66.67	73.12	83.33	58.82	68.97	20.83	29.41	24.39
BlackBerry Pearl	100.00	83.33	90.91	100.00	80.00	88.89	28.12	60.00	38.30
BlackBerry Storm	82.98	76.00	79.34	82.76	68.57	75.00	30.16	54.29	38.78
HTC Desire	100.00	94.44	97.14	100.00	95.83	97.87	26.09	50.00	34.29
HTC Dream	89.61	91.89	90.74	75.86	91.67	83.02	30.00	56.25	39.13
HTC Evo 4G	100.00	50.00	66.67	100.00	88.89	94.12	66.67	88.89	76.19
HTC Hero	90.48	100.00	95.00	92.31	100.00	96.00	15.62	41.67	22.73
HTC Magic	96.92	86.11	91.20	83.33	81.40	82.35	25.84	53.49	34.85
HTC Titan	100.00	100.00	100.00	90.00	100.00	94.74	10.00	33.33	15.38
HTC Touch	100.00	66.67	80.00	83.33	71.43	76.92	8.33	14.29	10.53
HTC Touch Diamond	100.00	94.12	96.97	90.48	95.00	92.68	14.89	35.00	20.90
HTC Touch HD	100.00	80.00	88.89	100.00	80.00	88.89	36.36	40.00	38.10
HTC Touch Pro	100.00	75.00	85.71	100.00	87.50	93.33	33.33	50.00	40.00
HTC TyTN	100.00	88.89	94.12	93.33	93.33	93.33	31.82	46.67	37.84
HTC TyTN II	100.00	91.67	95.65	88.24	93.75	90.91	16.28	43.75	23.73
HTC Universal	88.24	100.00	93.75	64.29	100.00	78.26	16.00	22.22	18.60
HTC Wizard	100.00	90.00	94.74	100.00	92.31	96.00	31.82	53.85	40.00
INQ Chat 3G	100.00	92.31	96.00	86.67	92.86	89.66	38.10	57.14	45.71
LG Dare (VX9700)	83.33	55.56	66.67	88.24	65.22	75.00	41.46	73.91	53.12
Average	95.39	83.30	**88.24**	89.59	86.13	**87.15**	27.46	47.59	**34.34**

Table 4.1: MUC score for Rule-based coreference resolution system for Wikipedia

4.6.2 Review domain

As described in Section 4.4.2, our gold standard featured manually annotated markables, whereas our algorithm for coreference resolution used automatically annotated markables. Therefore to compare gold standard and output of our algorithm we had to match manually annotated expressions and automatically identified ones. We decided to use heads of noun phrases to account for possible differences in the boundaries of manually and automatically annotated noun phrases. This was based on the fact that even if the boundaries of the NPs were different, we would be able to match NPs if they had the same head. We would consider our coreference algorithm's output correct if it identified "Black Iphone" as a part of the coreference chain and in our gold standard we found NP "amazing Black Iphone" with the matching head "Iphone". We used Machinese annotation and the labelled dependency links for the identification of the heads of the NPs. We believe that this introduced additional errors and it is discussed in more detail in the error analysis section (Section 4.7).

The baseline used for Wikipedia relied on the fact that texts discuss mobile phones and therefore subjects of the sentences are likely to be coreferential with the topic. However, the review described user experience and therefore had a lot of sentences where the subject was the first person singular pronoun. To account for this fact we have adapted the Wikipedia baseline: for the sentences where "I" was the subject, the object(s) were considered coreferential with the topic.

Table 4.2 presents the results of the evaluation for the review domain. In the

case of dealing only with definite noun phrases, our system can achieve 82.56% f-measure, whereas the baseline gets only 9.21% f-measure. However, when we also try to identify pronouns that are coreferential with the topic, f-measure drops to 71.73%. This issue is discussed in more detail in Section 4.7 which deals with error analysis.

	Rule-based: only NPs			Rule-based: with pronouns			Baseline		
File	**Precision**	**Recall**	**F-measure**	**Precision**	**Recall**	**F-measure**	**Precision**	**Recall**	**F-measure**
No Apple iPhone 3GS White	75.68	59.62	66.69	61.11	51.43	55.85	3.84	34.62	6.91
No Google Nexus S	100.00	83.33	90.91	68.52	65.52	66.98	5.57	56.67	10.15
No HTC Touch Touch Pro2	95.12	84.78	89.66	81.48	69.84	75.21	8.65	39.13	14.17
No HTC XV6900	91.67	85.19	88.31	75.76	83.87	79.61	7.03	33.33	11.61
No LG VX8575	89.47	80.95	85.00	89.29	75.76	81.97	5.15	33.33	8.92
No Motorola Droid	75.86	84.62	80.00	57.78	74.29	65.00	2.59	26.92	4.73
No Motorola Krave ZN4	81.13	79.63	80.37	72.00	78.26	75.00	14.29	51.85	22.40
No Nokia XpressMusic 5800	95.56	74.58	83.77	84.62	77.78	81.05	18.08	55.93	27.33
No RIM BlackBerry 9530	92.86	93.10	92.98	70.49	88.00	78.28	3.91	34.48	7.02
No Samsung Galaxy S	95.00	61.29	74.51	80.00	60.61	68.97	5.96	29.03	9.89
Yes Apple iPhone 3GS White	75.00	64.00	69.06	70.89	66.28	68.51	6.32	25.33	10.11
Yes HTC EVO 4G	93.75	78.95	85.71	75.00	75.00	75.00	1.70	18.42	3.11
Yes HTC G1	90.22	87.63	88.90	78.36	85.71	81.87	1.06	4.12	1.69
Yes HTC Touch Pro	87.50	84.91	86.18	60.47	79.71	68.77	5.80	35.85	9.99
Yes HTC Touch Touch Pro2	83.75	90.91	87.18	50.32	88.30	64.11	2.97	27.27	5.36
Yes Motorola Droid	82.50	87.18	84.78	62.32	86.27	72.37	3.92	30.77	6.96
Yes Nokia XpressMusic 5800	96.43	81.82	88.52	73.68	73.68	73.68	5.30	48.48	9.55
Yes Samsung Epic 4G	85.00	62.96	72.34	65.85	58.70	62.07	1.63	18.52	2.99
Yes Samsung Galaxy S II	70.00	56.76	62.69	51.06	58.54	54.55	2.08	18.92	3.74
Yes Sony Ericsson Z750a	95.08	92.06	93.55	81.19	91.11	85.86	4.44	26.98	7.62
Average	87.58	78.71	**82.56**	70.51	74.43	**71.73**	5.51	32.50	**9.21**

Table 4.2: MUC scores for rule-based coreference resolution system for the review domain

4.7 Error analysis

During the development of our method, several issues that affect the performance of the algorithm were identified. Inconsistencies in the annotation of the gold standard were a common problem for both kinds of texts, so these inconsistencies were identified and corrected. The results of the evaluation reported in the previous section were calculated after this correction was made. Other problems were specific to the kind of text in question and is discussed in the next sections.

4.7.1 Wikipedia texts

One of the problems we had to deal with when working with Wikipedia texts was caused by the contents of some articles which did not describe a model of a phone but the whole series of phones. In this case, the article did not have a main topic, but rather many subtopics. Given the fact that our experiment assumed the presence of the main topic, this kind of text was not processed correctly.

Automatic processing of the texts relies on the peculiarities we identified while studying the organisation of Wikipedia articles, e.g. it was noted that the first markable in the files denoted the topic. However this rule had exceptions and so the output of the system was incorrect in some cases. In order to address this issue we had to annotate manually the first expression which referred to the topic to ensure its correct identification.

Error analysis also showed that more deep linguistic processing of the texts should be involved, e.g. dependency relations should be taken into account in order to extract the second name of “HTC Magic” from the following example:

"The HTC Magic (known as the T-Mobile myTouch 3G in the US, and the docomo HT-03A in Japan) is a smartphone designed by HTC."

As mentioned earlier, the rule-based method is based on particularities in the way that names of submodels are formed and relations are expressed; however, exceptions result in low performance of the algorithm. It can be seen from the example of "HTC Touch": enhanced versions of the phone are called by completely different names - "the HTC P3452" and "the HTC P3050" and irregular ways of forming the name of the submodels drops the recall.

4.7.2 Review domain

As previously mentioned, the review texts were less technical and, therefore, the phone was less often referred to by its name. In addition, the reviewers were not using a standard name of the phone, but a rather general one, e.g., instead of using "Iphone 3gs" they would use "Iphone 3g" or even just "Iphone". These peculiarities resulted in the algorithm missing some mentions of the phone in the text.

The markables were identified automatically with the help of Machinese and this process introduced additional errors. First of all, the names of the phones were not always annotated as NPs, because they contained abbreviations or were missed by the algorithm, for example, "the 3GS". Therefore, at later stages of the algorithm's operation there was no way to link these mentions of the phone to the main topic.

Splitting a noun phrase into two shorter ones was another problem encountered

during the error analysis. There were cases when Machinese would not consider a name of the phone as constituting one noun phrase and therefore prevented the algorithm from linking this mention to the coreferential chain. A similar problem occurred when a noun phrase annotated by Machinese was shorter than the actual name of the phone; it would result in incorrect identification of the head of the noun phrase and poorer results in terms of the algorithm's output.

The evaluation results revealed difficulty in linking pronouns to the coferential chain. Given that we were interested in a high-precision algorithm, we had to take a very narrow window between the previous mention of the topic and the pronoun. It resulted in a lot of pronouns referring to the topic being missed and therefore a drop in the recall of the system. However, the attempt to make the window bigger resulted in the precision dropping considerably. Taking into account this information, it was decided not to process pronouns and to rely only on definite noun phrases.

4.8 Conclusions

This chapter described all stages of development of the coreference resolution for two types of corpora, including the elaboration of the annotation guidelines and the annotation of the corpora for coreference phenomena. It also presented a rule-based method for topic tracking for Wikipedia and review texts as well as its evaluation. The results of this approach are promising for most of the texts as it relies on the presence of a regular structure in the articles as well as special language used to talk about the product in question. Investigation of the texts

for which the performance is rather low revealed that even humans have problems analysing them.

The proposed method gave promising results. We have also proved that it is possible to use this method for different kinds of texts: both semi-structured texts from Wikipedia and unstructured texts which contain phone reviews. Therefore, we believe that the suggested approach can be generalised and used for other kinds of texts and not only Wikipedia and review texts. We also consider it possible to adapt it to other kinds of products.

This rule-based method for topic tracking will be used in the further stages of our research for identifying sentences containing mention of the phone and a phone feature linked to it. The next chapter (Chapter 5) will address the problem of identifying whether a feature mentioned in the same sentence as a phone is linked to that phone. This information will be used in order to select a subset of sentences from the corpus and use these for ranking features linked to the phone in question, which is described in Chapter 6.

Chapter 5

Annotation of links between phones and their features

5.1 Introduction

As previously described in Chapter 4, we are interested in extracting sentences that contain mention of a phone as well as a phone feature. However, only features related to the phone described in the review are relevant for us. Therefore given a sentence containing mention of the phone and phone features we need to identify whether these features belong to this phone.

To address this problem, a method inspired by relation extraction is employed. The next section (Section 5.2) provides a background in relation extraction and describes existing ways of approaching this task. In order to be able to benefit from the use of NLP techniques, we need to annotate the corpus and this step is discussed in Section 5.3. It is followed by Section 5.4, which discusses in more detail the methods employed to discover links between the phone and the features mentioned in the same sentence.

5.2 Relation extraction

As mentioned in Section 4.2.1, relation extraction (RE) is one of the steps of information extraction. It typically follows named entity recognition and coreference resolution and aims to gather relations between NEs. Culotta et al. (2006) define relation extraction as:

> "the task of discovering semantic connections between entities. In text, this usually amounts to examining pairs of entities in a document and determining (from local language cues) whether a relation exists between them."

Nowadays there are a lot of systems extracting relations from texts and there are different methods for dealing with this problem. Etzioni et al. (2008) classify all the methods used for relation extraction into three classes:

- knowledge-based methods;
- supervised methods;
- self-supervised methods.

Each of these classes are briefly explained in the remainder of this section.

5.2.1 Knowledge-based methods

The first category of methods is used usually in domain-specific tasks, where the texts are similar and a closed set of relations needs to be identified. Systems which use these methods rely on pattern-matching rules manually crafted for

each domain. However, not all the relations are domain-dependent and there are some domain-independent ones. Hearst (1992) describes the usage of lexico-syntactic patterns for extraction of hyponymy relations in an open domain. These patterns capture such hyponymy relations as between *"author"* and *"Shakespeare"*, *"wound"* and *"injury"*, *"England"* and *"European country"*. However, the author notes that this method does not work well for some other kinds of relations, for example, meronymy. This is explained by the fact that patterns do not tend to uniquely identify the given relation.

The systems which participated in MUC and deal with relation extraction also rely on rich rules for identifying relations (Fukumoto et al., 1998; Garigliano et al., 1998; Humphreys et al., 1998). Humphreys et al. (1998) mention that they tried to add only those rules which were (almost) certain never to generate errors in analysis; therefore, they had adopted a low recall and high precision approach. However, in this case, many relations may be missed due to the lack of unambiguous rules to extract them.

To conclude, knowledge-based methods are not easily portable to other domains and involve too much manual labour. However, they can be used effectively if the main aim is to get results quickly in well-defined domains and document collections.

5.2.2 Supervised methods

Supervised methods rely on a training set where domain-specific examples have been tagged. Such systems automatically learn extractors for relations by using machine-learning techniques. The main problem of using these methods is that the development of a suitably tagged corpus can take a lot of time and effort. On the other hand, these systems can be easily adapted to a different domain provided there is training data.

There are different ways that extractors can be learnt in order to solve the problem of supervised relation extraction: kernel methods (Zhao and Grishman, 2005; Bunescu and Mooney, 2006), logistic regression (Kambhatla, 2004), augmented parsing (Miller et al., 2000), Conditional Random Fields (CRF) (Culotta et al., 2006).

In RE in general and supervised RE in particular a lot of research was done for IS-A relations and extraction of taxonomies. Several resources were built based on collaboratively built Wikipedia (YAGO - (Suchanek et al., 2007); DBpedia - (Auer et al., 2007)). In general, Wikipedia is becoming more and more popular as a source for RE, e.g. (Ponzetto and Strube, 2007; Nguyen et al., 2007c,b,a). Query logs are also considered a valuable source of information for RE and their analysis is even argued to give better results than other suggested methods in the field (Paşca, 2007, 2009).

5.2.2.1 Weakly-supervised methods

Some supervised systems also use bootstrapping to make construction of the training data easier. These methods are also sometimes referred to as "weakly-supervised information extraction". Brin (1998) describes the *DIPRE* (Dual Iterative Pattern Relation Expansion) method used for identifying authors of the books. It uses an initial small set of seeds or a set of hand-constructed extraction patterns to begin the training process. After the occurrences of needed information are found, they are further used for recognition of new patterns. Regardless of how promising bootstrapping can seem, error propagation becomes a serious problem: mistakes in extraction at the initial stages generate more mistakes at later stages and decrease the accuracy of the extraction process. For example, errors that expand to named entity recognition, e.g. extracting incomplete proper names, result in choosing incorrect seeds for the next step of bootstrapping. Another problem that can occur is that of semantic drift. This happens when senses of the words are not taken into account and therefore each iteration results in a move from the original meaning. Some researchers (Kozareva and Hovy, 2010; Hovy et al., 2009; Kozareva et al., 2008) have suggested ways to avoid this problem and enhance the performance of this method by using doubly-anchored patterns (which include both the class name and a class member) as well as graph structures. Such patterns have two anchor seed positions "{type} such as {seed} and *" and also one open position for the terms to be learnt, for example, pattern "Presidents such as Ford and {X}" can be used to learn names of the presidents. Graphs are used for

storing information about patterns, found words and links to entities they helped to find. This data is further used for calculating popularity and productivity of the candidate words. This approach helps to enhance the accuracy of bootstrapping and to find high-quality information using only a few seeds.

5.2.3 Self-supervised systems

Self-supervised systems go further in making the process of information extraction unsupervised. The KnowItAll Web IE system (Etzioni et al., 2005), an example of a self-supervised system, learns "to label its own training examples using only a small set of domain-independent extraction patterns". It uses a set of generic patterns to automatically instantiate relation-specific extraction rules and then learns domain-specific extraction rules and the whole process is repeated iteratively.

The Intelligence in Wikipedia (IWP) project (Weld et al., 2008) is another example of a self-supervised system. It bootstraps from the Wikipedia corpus, exploiting the fact that each article corresponds to a primary object and that many articles contain infoboxes. This system is able to use Wikipedia infoboxes as a starting point for training the classifiers for the page type. IWP trains extractors for the various attributes and they can later be used for extracting information from general Web pages. The disadvantage of IWP is that the amount of relations described in Wikipedia infoboxes is limited and so not all relations can be extracted using this method.

Etzioni et al. (2008) introduce the notion of Open Information Extraction,

which is opposed to Traditional Relation Extraction. Open information extraction is "a novel extraction paradigm that tackles an unbounded number of relations". This method does not presuppose a predefined set of relations and is targeted at all relations that can be extracted.

All the methods described above have advantages and disadvantages and the choice depends greatly on the task in mind and the accuracy needed. For our research we adopt a supervised approach to relation extraction and more details are provided in Section 5.4.

5.3 Corpus annotation

As previously mentioned, we are interested in identifying whether a mobile phone has a link to the features mentioned nearby. It should be emphasised that we consider only cases where a mention of the phone and the phone's features are in the same sentence. Boundaries of sentences were taken into account rather than clauses. After the coreferential links are resolved we can get a chain identifying the topic of the review (mobile phone in our case). At the next stage we can attempt to identify whether there is a link between the phone and the feature mentioned. However, in order to develop an automatic method for identifying links between phones and their features, we first need to obtain data and annotate the corpus for this phenomenon. Given that the review corpus was bigger than the Wikipedia one, it was decided to use primarily review texts for finding the best ranking of features. Therefore, we aimed to develop relation extraction using the review corpus. However, we believe that the suggested approach can be generalised and

used for other kinds of texts.

Thus, the review corpus was annotated for links between the mentions of the phones and features. The same annotation tool, PALinkA, as described in Section 4.4.1, was used for the annotation of the links between mentions of the phone and features in the same sentence. Initially, we automatically pre-annotated features in the review corpus and the annotator was presented with texts which had some features already explicitly marked. After this, the annotation of features consisted of several steps: pre-annotating the features, marking irrelevant features, annotating features missed by the automatic method and linking features and the phone. Therefore, the annotator had the option to keep the feature identified during the pre-annotation, mark it as incorrect, or annotate new features. This information was later used for evaluation of the pre-annotation which is described in Section 5.3.2.

At the next step the annotator had to identify cases where features and a phone were mentioned in the same sentence and mark whether there was a link between those two items.

This annotation helped us to collect 984 instances of link relation and this information was used for the relation extraction which is described in Section 5.4.

5.3.1 Pre-annotation of features

As previously mentioned, in order to make the annotation easier and more consistent, we pre-annotated the corpus of the review texts with the features. We are also looking for ways to extract information in a semi-automatic manner, so if this method proves precise enough, we can attempt to annotate features automatically in the future without any human input.

We used the list of features described in Section 3.2 for the pre-annotation. This list was collected using infoboxes describing mobile phones and featured 4720 items, which consisted of both features and their values. It was decided to treat features and values as equal on account of value actually referring to the corresponding feature. We used it to assist the annotation and also to see how precisely infoboxes describe the features mentioned in the text. Given that we want to use infobox features afterwards, it was also important to take this list as a base for our annotation. The use of an automatically compiled list gives us the possibility to adapt the employed methodology to other domains, which is an important aspect of our research.

The features are identified in a text in a greedy manner. This means that during the pre-annotation, longer sequences of features are preferred. For example, if we had "qwerty" and "keyboard" as separate features and at the same time a different feature "qwerty keyboard", we would link this sequence to the longer feature, "qwerty keyboard".

Statistics of the number of features annotated in a given text were collected

and used at the later stage to choose files for the annotation. It has previously been discussed that the review corpus consisted of 3,392 texts and a method to choose files for annotation was needed. For this reason, we decided to choose files that had the largest amount of annotated features. This should guarantee that we can get as many examples of the usage of a feature as possible and evaluate the pre-annotation effectively. Therefore, a total of 40 files were annotated for links between the phone and phone features.

The next section provides some further details about the evaluation of the pre-annotation.

5.3.2 Evaluation

In order to evaluate how accurate the pre-annotation process is, we asked the annotator to mark the features with a special tag if he/she considered them as incorrect. It helped us to estimate how many features were incorrect and how precise the pre-annotation was. We also attempted to evaluate the coverage of the pre-annotation: for this purpose the annotator was asked to identify features that he/she deemed important and that were missed by the automatic pre-annotation.

A total of 20 files (almost 77,500 words) were annotated with these specially introduced tags (for incorrect items and missed features). These files contained 4602 pre-annotated features in total. The evaluation showed that 1466 features out of 4602 were deemed incorrect by the annotator, which gives us 68.14% precision. The annotator also tagged an additional 1673 features, and therefore the recall of our method is 65.2%. Lists from Wikipedia proved to be not as precise as it was

hoped; however, it is discussed in more detail in the next section.

For this part of the corpus we also collected statistics about the number of links annotated. If we take into account that out of 4602 pre-annotated features 1466 items are incorrect and also 1673 features are missed by the automatic pre-annotation, we will obtain a list of 4809 correct features annotated in the corpus. Out of these 4809 features, only 648 (13.5%) were encountered in the same sentence as some phone and were linked to it.

Statistics about the links revealed that more texts need to be annotated in order to get more cases of annotated links between a feature and phone. A bigger corpus was needed to be able to use machine learning, so we went on annotating but focused only on annotating features which were in the same sentence with some mention of the phone, not taking into account other occurrences of the features. An additional 20 files were annotated. Therefore, the corpus which is used in Section 5.4 for identification of links between features and the model of the phone consists of 984 links (a total of 40 files).

5.3.3 Problems with annotation

Several problems were encountered during the annotation of the features and links. At first, we discovered that the pre-annotation of the features was not very precise, as noted in the previous section. Some of the features from our list were too general and therefore ambiguous. For example, the list of features contained some items as “left”, “right”, “active”, etc., which are obviously values of some features. However, these items are ambiguous and rarely act as actual values of the features:

in most cases this will be just a general adjective.

Another problem we encountered was that the phone and the features were rarely mentioned in the same sentence. In most cases features were mentioned on their own and even when appearing in the same sentence they were often quite distant from the phone. Although we annotated links in terms of the sentence, we feel that in some cases it would be difficult for automatic methods to judge the presence of the links, because the items are too far away from each other.

In some cases it was quite difficult for the annotator to decide whether there was a direct link between the feature and the phone, because it was not lexically, explicitly expressed. For example, "the choice between the htc evo 4g and the samsung epic 4g would be difficult unless you have feelings regarding a physical keyboard , or the custom skin". In this respect, cases like "this phone has a 5.0 mpx camera" were much easier for the annotator. However, it was noticed that the presence of lexical cues depended heavily on the personal writing style of the review authors. Some of the reviewers were using very explicit language, which had a lot of indications of whether the feature belonged to the phone or not and therefore these texts were easier to annotate. We will try to take this into account when developing automatic identification of the links, which is described in the next section.

5.4 An automatic method for relation extraction

We initially approached the relation extraction by deciding which methods would be the best to solve this problem. After examining the data, we found out that

we cannot simply rely on patterns which can be identified using hand-crafted rules and additionally, that there are a lot of factors that affect the presence of the links between mentions of the phones and phone characteristics. Therefore constructing a rule-based system for identification of these links could be too costly and inefficient. Moreover, as we aim to develop methods which can be easily adapted to other domains, the use of a rule-based system does not seem an appropriate way to tackle the problem.

Machine learning allows for the combination of a lot of different predictors, features that indicate that there is a link. It gives an opportunity to learn from the data already collected (manually-annotated gold standard) and also to see which features are the most useful. Whilst we anticipate that the stage of error analysis is more difficult for a machine-learning approach than for a rule-based one, taking into account the nature of the data and the described benefits of machine learning, we decided to adopt this approach for our task.

More details about the machine-learning approaches are provided in the next section (Section 5.4.1). Section 5.4.2 describes the features used for machine learning, and is followed by Section 5.4.3 which presents the results of the developed method. Section 5.4.4 discusses the results and also focuses on the error analysis.

5.4.1 Machine learning

Machine learning can refer both to the branch of artificial intelligence and the methods used in this field. Overall, if talking about the latter, machine learning can be defined as improving performance in some task with experience (Mitchell, 1997).

However, this definition is quite a broad one, with a more specific description stating that machine learning deals with systems that can learn from data. The initial data, called *training data*, is used to tune the initial model that can be used afterwards on unseen data, typically referred to as a *test set* (Bishop, 2006). Generalisation of the model that has been learnt allows it to be used for new examples that were not previously processed by the algorithm. As Bishop (2006) mentions, in most applications the input data is preprocessed to produce a new representation, which will describe this data in terms of a set of variables. This process is typically called *feature extraction*[1]. It should be noted that the same process of feature extraction is used for both training and test sets.

All machine learning methods can be classified in several categories: supervised learning, unsupervised learning and reinforcement learning (Bishop, 2006). Supervised learning methods acquire knowledge from data that has been explicitly annotated with category labels or structural information (Mooney, 2003). This is usually done by human annotators to ensure a high quality of data. In unsupervised settings there is no expert human annotation and unlabelled examples are clustered based on similarity into coherent groups. These algorithms take as input a set of features representing each example without any corresponding target value. These methods can be used when it is too difficult to acquire a set of labelled examples for the task. The last category of machine learning methods, reinforcement learning, is learning how to solve a problem that is, which actions

[1]The term "feature" is employed both for phones and ML. However, we will make sure to explicitly state which kind of feature we are talking about in cases where there is ambiguity. When it is difficult to make a difference between phone feature and ML feature, the term "phone characteristic" will be used to refer to the former one.

to take in a given situation in order to maximise a numerical reward (Sutton and Barto, 1998). Unlike supervised algorithms, reinforcement learning does not get labelled examples as input, but learns the best behaviour from its own experience.

In this thesis, annotated data is available to us; therefore, our task is suitable for supervised machine learning. We have a classification problem where we need to find the most probable label for a new example given the training data we have seen. The result of classification systems is typically evaluated in terms of their precision, the percentage of examples the model was able to classify correctly. As discussed in (Mooney, 2003), to guarantee that the precision results are not biased to the splitting of data in training and test sets, data is split (into training and test sets) in various ways, and afterwards the average precision of these splits is taken. It ensures objective evaluation and also provides information about the performance variation of the algorithm.

There are a lot of different learning algorithms which can be used for this kind of problem, but we have chosen the Naive Bayes classifier. This algorithm is considered among the most effective algorithms known for classification tasks (Mitchell, 1997). It is a simple probabilistic classifier based on applying Bayes' theorem with strong (naive) independence assumptions. It assumes that all features are independent from each other, i.e. the presence (or absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature. The Naive Bayes classifier requires a small amount of training data to estimate the parameters, i.e. to train the model, which is necessary for classification. Another reason for choosing the Naive Bayes classifier was the fact

that it is already implemented in the linguistics library NLTK (Bird et al., 2009) for Python, which was used forthe processing of all data in this thesis.

The next section describes in more detail the feature extraction which was used for training machine learning.

5.4.2 Features

We need to train a machine learning algorithm that takes as input a sentence with an explicitly marked mention of the phone and phone characteristic and determines whether this characteristic belongs to the phone or not. We started feature engineering by examining the texts and trying to identify what can indicate the presence of a link between the mention of the phone and its characteristics (features).

As a starting point we used research by Zhou et al. (2005) who explore various features used for the relation extraction. We examined the features suggested in this paper and selected the ones relevant to our task. We modified some of the features to fit our research setting and data. It resulted in the following list of features: bag of words, filtered bag of words, context, filtered context, distance between mention of the phone and the phone feature, overlap between mentions of the phone and the phone features. They are discussed in more detail later in this section.

We also considered work by Chan and Roth (2011) indicating that the use of syntactico-semantic structures can be beneficial for the relation extraction. The authors mention five different syntactico-semantic structures: premodifying,

possessive, preposition, formulaic and verbal. However, we considered only two of them relevant to our task: possessive and preposition. Therefore features indicating the presence of possessive relations and prepositions were added.

We have added several more machine learning features that were inspired by studying our data: presence of indicating phrase, name of the phone feature, head of the phone feature, whether the mention of the phone and the phone features depend on the same verb.

All features can be further grouped into several categories: bag of words, name of the phone characteristic, indicating phrases, distance and syntactico-semantic relations. The following sections describe these classes of features in more detail.

5.4.2.1 Bag of words

We employed four variations of bag of words features in our research:

1. bag of words (bow);
2. filtered bow (without stop words and/or punctuation);
3. context;
4. filtered context (without stop words and/or punctuation).

The first feature was a standard version of bag of words, which is typically used in text classification. The occurrence of each word in the sentence is used as a feature: if the word is present, the feature gets the name of this word and value "TRUE". We also tried using a filtered bag of words where we would ignore stop words and/or words that are less than two characters long.

At the next stage we decided to impose a restriction on which words should be used as features, and therefore we looked at the context of the phone characteristic. Various windows were used, ranging from two to five words; only 2-5 words that were before and after the phone characteristic were considered as features. We also used a variation of the context and filtered all stop words and/or words that are less than two characters long.

5.4.2.2 Name of the phone characteristic

Another class of features deals with the phone characteristic involved in the relation. We considered using two different features:

1. name of the phone characteristic;
2. head of the phone characteristic.

We extracted the name of the phone characteristic under consideration and presented features in the following way: ('VALUE' : 'name of the characteristic'). We also considered adding a feature which represents the head of the phone characteristic, e.g. if the phone characteristic is "phone memory", we would generate a feature ('HEAD_VALUE' : 'memory'). This feature was added in order to see whether further generalisation helps to improve machine leaning.

5.4.2.3 Indicating phrases

The next feature that was considered was presence of indicating phrase in the sentence. We were looking for phrases that may indicate that the characteristic

of the phone and the mention of the phone are linked. The following indicating phrases were taken into account:

- “has”;
- “comes with”;
- “supports”;
- “includes”.

A further restriction was also imposed on the extraction of the indicating phrases: the expression describing the phone preceded the phone characteristic and the gap between them was no more than four words.

5.4.2.4 Distance

We explored several variations of the distance feature:

1. distance between the mention of the phone and the phone characteristic (in tokens);
2. distance between the mention of the phone and the phone characteristic (in tokens, based on thresholds);
3. whether there is a token overlap between the mention of the phone and the phone characteristic.

First, we extracted the distance between the mention of the phone and the phone characteristic in tokens. Thus, our feature “DISTANCE” had a numeric value for each sentence.

The next step was to calculate an average distance between the mention of the phone and the phone characteristic in the whole corpus, which turned out to be nine tokens. This value was then used to implement a distance feature based on thresholds. If the distance was less than or equal to average distance, the calculated feature would get the value "TRUE", and if more - "FALSE".

The last feature of this group was checking whether there is an overlap of tokens belonging to the phone and those belonging to the phone characteristics. If there was an overlap, then feature "OVERLAP" would get value "TRUE".

5.4.2.5 Syntactico-semantic relations

Syntactico-semantic relations were the last group of features used for machine learning:

1. whether there is a possessive relation between the mention of the phone and the phone characteristic;
2. whether there is a prepositional relation between the mention of the phone and the phone characteristic;
3. whether the mention of the phone and the phone characteristic are linked via the verb in the parse tree.

The first possible indicator of the relation was presence of possessive relation. In order to extract it, we verified that the mention of the phone preceded the phone characteristic and also that the gap between them was no more than two tokens. If it was valid, we checked whether any of the tokens in the expression denoting

phone had value "GEN" as a part of "morpho" tag (here, Machinese markup was used). This allowed us to find out whether there was a possessive relation.

To find prepositional relations, we used a similar approach to the one employed for the extraction of possessive relations: we checked that the phone characteristic preceded the mention of the phone, that the gap is no more than three tokens, and that one of the tokens between the phone characteristic and mention of the phone has the value "PREP" in "morpho" tag (i.e. is a preposition).

We employed the parse tree to find out whether the mention of the phone and the phone characteristic was linked over a common verb (i.e. there was a path in the dependency tree linking them via the verb). It was observed that in order to get more precise results we only needed to check depth two: we tested whether heads of the noun phrases representing the phone and the phone characteristic depended on the same verb or whether there was a link to the verb via one other element.

5.4.3 Evaluation

Once the training using the features described in the previous section was done, we proceeded to evaluate the performance of different combinations of the features. Our corpus consisted of 1991 sentences containing a mention of a phone and a phone characteristic linked to it. For evaluation purposes, we used 5-fold cross-validation, where the initial annotated corpus was randomly split into five equal-size parts. At each iteration, four folds were used for training and one fold for testing. The final result of the evaluation of machine learning is the average over

Method	**Precision**
Bag of words (BOW)	**59.67**
BOW (punctation and less than 1 character words filtered)	59.67
BOW (stop words filtered)	59.21
BOW (stop words and punctuation filtered)	58.96
Context of the phone feature (window = 2)	59.42
Context of the phone feature (window = 3)	**61.08**
Context of the phone feature (window = 4)	59.27
Context of the phone feature (window = 5)	58.96
Context (window = 2) filtered for punctuation and stop words	57.91
Context (window = 2) filtered for stop words	58.06
Context (window = 2) filtered for punctuation	59.47
Context (window = 3) filtered for punctuation and stop words	59.17
Context (window = 3) filtered for stop words	59.47
Context (window = 3) filtered for punctuation	60.47
Context (window = 4) filtered for punctuation and stop words	59.27
Context (window = 4) filtered for stop words	59.47
Context (window = 4) filtered for punctuation	59.82

Table 5.1: The evaluation results for bag of words

all five results, acquired with different ways of splitting the initial data.

Different variations of bag of words were chosen as a baseline, the results are shown in Table 5.1. The results revealed that simply using all the words in the sentence is one of the most effective methods; it gets 59.67% accuracy. Only three-word context of the phone characteristic gets considerably better results than simple bag of words - 61.08% accuracy.

At the next stage of our research, we tried using different combinations of the ML features described in Section 5.4.2. The results can be found in Table 5.2.

We tried to combine our features with bag of words; however, it resulted in a considerable drop in the performance of the method. As it can be seen from the Table 5.2, the best performance (73.28% accuracy) was acquired when dropping the “overlap” feature. However, further removal of the features reduces

Method	Precision
[1] Name of the feature	57.76
[2] Head of the name of the feature	64.74
[3] Indicating phrase	54.90
[4] Distance	64.84
[5] Distance (threshold = 5)	55.15
[6] Distance (threshold = 8)	57.16
[7] Distance (threshold = 9)	57.41
[8] Distance (threshold = 10)	56.50
[9] Overlap of the phone and phone feature	55.75
[10] Possessive relation	53.29
[11] Prepositional relation	55.45
[12] Verb dependency	55.40
[13] Verb dependency (2 levels up)	57.81
[4] + [7]	65.44
[1]+[3]+[4]+[6]+[7]+[8]+[9]	71.17
[2]+[3]+[4]+[6]+[7]+[8]+[9]	71.52
[2]+[3]+[4]+[7]+[9]+[10]+[11]+[13]	72.02
[1]+[2]+[3]+[4]+[7]+[9]+[10]+[11]+[13]	73.18
[1]+[2]+[3]+[4]+[10]+[11]+[13]	72.32
[1]+[2]+[3]+[4]+[7]+[10]+[11]+[13]	**73.28**

Table 5.2: The evaluation results for all feature combinations

the performance.

This section described the results of evaluation and the next section discusses them further.

5.4.4 Error analysis

Machine learning implemented in NLTK allowed us to explore the most informative features that were used by the algorithm. Table 5.3 shows which features were the most informative for machine learning. It can be seen that a longer distance such as 18-19 tokens between the phone and the phone characteristic strongly indicates absence of relation, whereas the proximity of two items signifies that they most probably are linked. Prepositional relation is also a strong indicator of the link

	Feature	**Ratio**
1.	DISTANCE = 19	no_lin : link = 5.0 : 1.0
2.	indicating_phrase = True	link : no_lin = 4.7 : 1.0
3.	PREP = True	link : no_lin = 4.7 : 1.0
4.	VALUE = 'usb'	link : no_lin = 4.0 : 1.0
5.	VALUE = 'sprint'	no_lin : link = 4.0 : 1.0
6.	HEAD_VALUE = 'sprint'	no_lin : link = 3.9 : 1.0
7.	DISTANCE = 18	no_lin : link = 3.8 : 1.0
8.	HEAD_VALUE = 'button'	link : no_lin = 3.3 : 1.0
9.	HEAD_VALUE = 'card'	link : no_lin = 3.3 : 1.0
10.	HEAD_VALUE = 'headset'	link : no_lin = 3.3 : 1.0
11.	HEAD_VALUE = 'life'	link : no_lin = 3.3 : 1.0
12.	VALUE = 'battery life'	link : no_lin = 3.3 : 1.0
13.	VALUE = 'samsung'	no_lin : link = 2.7 : 1.0
14.	VALUE = 'keyboard'	no_lin : link = 2.7 : 1.0
15.	VALUE = 'at'	no_lin : link = 2.7 : 1.0
16.	HEAD_VALUE = 'samsung'	no_lin : link = 2.7 : 1.0
17.	HEAD_VALUE = 'android'	no_lin : link = 2.7 : 1.0
19.	HEAD_VALUE = 'at'	no_lin : link = 2.7 : 1.0
20.	HEAD_VALUE = 'flash'	link : no_lin = 2.6 : 1.0
21.	VALUE = 'headset'	link : no_lin = 2.6 : 1.0
22.	VALUE = 'mp3'	no_lin : link = 2.4 : 1.0
23.	HEAD_VALUE = 'mp3'	no_lin : link = 2.4 : 1.0
24.	DISTANCE = 3	link : no_lin = 2.3 : 1.0
25.	VALUE = 'android'	no_lin : link = 2.1 : 1.0

Table 5.3: Top 25 most informative features

between the phone and its feature.

The evaluation section showed that our methods outperform baseline by more than 10%, but it also revealed that there are cases where links were misclassified. We carried out error analysis in order to find out the reasons for the mistakes and, therefore, how we can enhance our algorithm.

When analysing incorrectly classified links between the phone and the phone characteristics, we discovered the following major problems: incorrect annotation, ambiguous cases, the characteristic being too far away, and the algorithm relying too heavily on VALUE. The annotation task is a difficult one and in some cases

the correct links were missed by the annotator. It resulted, for example, in the algorithm finding a relation where it was not annotated initially. For example, the sentence "the internal display of the vx8700 is very nice even under varying light conditions" features an obvious link between "display" and "the vx8700"; however, the initial annotation had missed it.

Another problem the algorithm encountered was the mention of the phone and phone characteristic are too far apart in the sentence. As distance is a strong indicator, there is a high chance that if the distance exceeds the average, the sentence will be classified as having no link. One of the problems here is the presence of long enumeration lists such as in the sentence, "the phone includes calendar , calculator , alarm , world clock , notepad and tip calculator tools", where the last element is very far away from the mention of the phone. Therefore, the algorithm should be enhanced to capture enumeration lists and calculate distance in these cases in a different manner.

Some cases are ambiguous and it is difficult to decide whether there is a relation or not, for example, "the lg vx8700 has a 2 mega pixel camera which also takes video". We can attribute feature "video" to the phone itself or only to the camera. Therefore, even for human annotators, it is difficult to decide on the presence or absence of the relation.

More sophisticated algorithms are needed to treat, for example, negation in order to correctly classify cases like "i believe that there is a way to actually upload music to the phone using the not included usb cable , but i have not yet tested this as i have no such cable".

5.5 Conclusions

This chapter addressed the problem of relation extraction and described relevant research, corpus annotation and also an automatic method used to annotate links between mentions of phone and features. We described in detail the features used for our algorithm and the way they were extracted. The 5-fold cross-validation of different combinations of features was also presented.

Bag of words and its various modifications were considered baseline for our method; the best baseline achieved 61.08% accuracy. The best combination of the features for our machine learning achieved 73.28% accuracy. It was obtained by using the name of the feature, head of the name of the feature, indicating phrase, distance, distance (threshold = 9), possessive relation, prepositional relation and verb dependency (two levels up).

We also examined the errors in the classification output and attempted to identify the reason for the algorithm failing to classify sentences correctly. Several suggestions of further possible directions for enhancing the algorithm were suggested as well.

The results of this machine learning algorithm will be used in the next chapter (Chapter 6) for limiting the number of sentences to be processed by the algorithm and in order to remove noise.

Chapter 6

Ranking

6.1 Introduction

As mentioned in Chapter 1, we are exploring the ways to design an interactive question answering system that will assist users in choosing a product in an optimal way. It is especially important when a large number of similar products are available. Such an IQA system will be based on selecting a set of characteristics (also referred to as product features) that describe the relevant product, and in this way narrowing the search space. This approach treats IQA as a problem of constraint management, similar to those described in Section 2.6.1. Each phone feature is regarded as a constraint and we believe that the order in which these constraints are presented in terms of IQA sessions is of high importance. Therefore, they need to be ranked in order to have a dialogue which selects the product in an efficient manner.

One of the hypotheses explored in this research is whether product characteristics mentioned in user reviews are important for a person who is likely to purchase a product and can therefore be used when designing an IQA system. We propose a corpus-based method for weighting the importance of product features using a corpus of reviews. Our assumption is that these texts will focus on

the features that are more important for users and therefore are more likely to determine the purchase of the product. We are exploring different methods to rank product features in order to be able to provide an IQA system with information on which product characteristics should be given priority and presented first.

The chapter is structured as follows: The next section (Section 6.2) discusses the related work in the field. Section 6.3 presents a description of the experiment including its justification and ranking methods developed. Two types of evaluation are carried out: intrinsic (Section 6.4) and extrinsic evaluation (Section 6.5). Finally, the results are discussed in Section 6.6.

6.2 Related Work

We address the problem of content management for interactive question answering systems which is related to dialogue managers that constitute a part of dialogue systems. However, to the best of our knowledge, there is no study similar to the one carried out in this chapter. In addition, our approach is novel because it lies at the intersection of several NLP fields such as information extraction, IQA and sentiment analysis. For this reason, work in information extraction, sentiment analysis and interactive question answering can be considered as the most relevant to our research and is briefly presented next.

There are a number of projects focusing on extraction of product features for sentiment analysis. The system described in (Hu and Liu, 2004) extracts opinion summaries about products, but instead of getting the opinion about the product in general, the proposed method tries to produce an opinion summary about separate

product features. For this purpose, it mines product features discussed by the customers and rates each opinion as positive or negative. This information is later used to produce feature-based summaries about the products. Meng and Wang (2009) aim to solve a similar problem but use multiple specifications of a product for further clustering and extracting of product features. It is also done in order to produce summaries describing the products. Opine (Popescu and Etzioni, 2005) is an example of an unsupervised information-extraction system which mines reviews in order to build a model of important product features. Its output also describes opinions of the reviewers about different product features and their relative quality across various products.

Different approaches were developed to address the problem of extraction of product characteristics: unsupervised (Raju et al., 2009) and semi-supervised methods (Zhai et al., 2011a), as well as topic modelling (Zhai et al., 2011b). Some researchers attempted to build specialised domain ontologies manually in order to get better quality resources, but we are aware of only one ontology describing mobile phones (Junwu et al., 2010).

Our research differs from the aforementioned works, because we do not focus on extracting features of the phone from the reviews. Instead, we are more interested in ranking already acquired lists of features using the available customer reviews. In this respect, and keeping in mind the goal of our research, it is worth mentioning work previously done in the field of IQA.

Although we are not aware of applications in the field of IQA similar to our research, there are several IQA systems that address the problem of effective

information management, which can be considered relevant to the work in question. These systems attempt to help users choose products and rely on constraint-based approaches (Qu and Green, 2002; Varges et al., 2007; Rieser and Lemon, 2009). They were presented in more detail in Chapter 2.

These systems focus on the interaction when a constraint-based approach is used, but none of them try to rank the constraints or propose methods to make search for information more optimal in this way. In all the cases, either hand-crafted or learnt policies are used to decide which dialogue move to take next. These systems try to act according to the number of results they retrieve, and on the basis of this information they attempt to relax the request or ask for additional constraints. We are more interested in suggesting new constraints to the customer and would like to select those that will help the user to choose a product in the quickest time. This aspect of the problem is not discussed in these research works.

6.3 Experiment

Given that one of the aims of this thesis is to optimise the process of selecting a product on the basis of its features using an IQA, we evaluated several methods for ranking features. These methods are presented later on in this section, and the obtained rankings are evaluated in Section 6.4 and Section 6.5. We start this section by providing a justification for the experiment.

6.3.1 Justification of the experiment

One way to identify the importance a feature has in terms of choosing a product is to collect and analyse a large number of interactions between a human and a sales assistant or a computer. This information can be then used to learn the appropriate ranking of the features to be presented to the user. However, this approach is labour-intensive and time-consuming, which makes it very expensive, especially because of its domain dependency. As a result of the domain dependency, information gathering needs to be repeated every time a system is adapted for a new domain. For this reason, we propose a method which relies on user reviews to determine the ranking of the product features.

The underlying assumption of this method is that the most important features will also be mentioned frequently in the user reviews. Therefore, we believe it is possible to propose several weighting schemes which take a corpus of reviews and produce the ranking. Given that these reviews contain a large number of opinionated sentences, NLP techniques are being used to differentiate between positive and negative sentences. This is done in order to identify whether certain types of sentence (e.g. positive) are more likely to contain the necessary information to rank the product features correctly.

6.3.2 Ranking methods

We developed several methods for the ranking of product features on the basis of their occurrence in our corpora. In this thesis, features of mobile phones are examined, but the method can be adapted to other products if data is available.

To carry out the experiments, we had to first identify features that could be of interest to users and therefore need to be ranked. Manual construction of such a list did not seem objective enough and therefore we relied on semi-automatic methods, described in Section 3.2. It allowed us to get a list of features based on Wikipedia infoboxes; values corresponding to the features were also collected as a way of identifying indirect references to the features in the text.

6.3.2.1 Ways to match features

Once the list of features had been collected, we were able to investigate ways of ranking them. As previously mentioned, it was decided to use NLP techniques to find the best ranking algorithm.

Given the fact that a product feature can be expressed in several ways, we employed several methods for matching the features extracted from the Wikipedia infoboxes with their occurrences in the texts. For all the ranking methods described in the next subsection, three types of matching methods were used:

- surface-based (also referred to as strict match),
- fuzzy matching (e.g. *battery life* and *lifespan*),
- values for features (e.g. *5 megapixels* and *camera*).

Surface matching implies a strict match between the string denoting a feature from the Wikipedia infoboxes and a string in the corpus. This matching technique does not allow any flexibility in how the feature is expressed in the text. Therefore, this type of matching brings some limitations, as language is ambiguous and there are many ways to express the same thing using different surface representations. For this reason, we also implemented a *fuzzy matching* method which takes into consideration not only the surface form, but also considers synonyms extracted from WordNet (Fellbaum, 1998) and manually compiled lists. Several of the problems identified with the first method were solved using fuzzy matching and are discussed in Section 6.4.6. At the same time, fuzzy matching introduces its own errors which are discussed in the same section.

Another way to improve the matching algorithms is to consider that a feature occurs in a text not only when it is directly mentioned, but also when values corresponding to a feature are used. Despite the appeal of this approach, there are values which are multiword expressions, so strict matching would give a very low recall. For this reason, we used heuristics which consider a match successful if at least 60% of the text denoting a value was found. This helped us identify more information, but revealed the problem of overlapping features, which is further discussed in Section 6.4.6.

6.3.2.2 Frequency-based ranking

The first method explored relies on the frequency of a feature in our corpus of reviews in order to determine its importance. The assumption here is that the

more frequently a feature is mentioned, the more important it is for the users. This approach was inspired by automatic summarisation (Luhn, 1958). Therefore, we extract the frequency of each particular feature mentioned and use it as its score. For this purpose, all three different types of matching mentioned in the previous subsection are used.

6.3.2.3 Opinion-based ranking

Given that we are dealing with a corpus of reviews, we thought it could be beneficial to use the polarity of the sentences contained in the reviews in the ranking process. In order to investigate this, a sentiment classification system is also employed to distinguish between features mentioned in positive and negative contexts. For determining the polarity of a sentence, we use a lexicon-based algorithm based on the SO-CAL algorithm (Taboada et al., 2011). This method relies on a dictionary containing words and their semantic orientation scores related to the sentiment expressed. This semantic orientation ranges from -4 to 4, where -4 stands for a totally negative word and 4 for a totally positive word. For our experiments, we use the dictionary developed for the original method (Taboada et al., 2006).

In the above-mentioned method, the polarity of a sentence is measured as the sum of the semantic orientations present in the words. These words and their POS tags are checked in the dictionary in order to find out their semantic orientation. Negation markers, modals and intensifiers change the polarity for the next word. The sentence is labelled as positive or negative if the overall semantic orientation is positive or negative, whereas sentences with score 0 are labelled as being neutral.

We developed two ranking methods based on the identification of the opinion in the text. The first of them takes into account only opinionated sentences and ignores the neutral ones. The assumption is that the authors of the reviews will express opinions (positive or negative) about the features important to them; therefore, frequency-based ranking can be applied to the sentences that contain sentiment information in order to get a reliable ranking. However, we should mention that we did not attach opinions to the particular features and just identified them at the sentence level.

Taking into consideration that neutral sentences may contain information that could be useful for the ranking, a weighted ranking method which relies on opinion information was implemented. In this method, each occurrence of a feature in a neutral sentence receives a score of only 0.5, whereas an occurrence in an opinionated sentence gets a score of 1. In addition, two more experiments were run which considered only the positive and negative sentences for computing the ranking.

6.4 Intrinsic Evaluation

6.4.1 Corpus description

For the experiments reported in this chapter, two corpora were used: Wikipedia articles describing mobile phones and a corpus of reviews from the Epinions.com website (both corpora were described in more detail in Section 3.1). User reviews corpus consisted of 3,392 reviews (114,708 sentences; 2,253,877 words) organised into two labels: “yes” and “no”. However, these labels reflecting the user’s opinion

about the product in general were not used for the ranking at this stage.

One of the hypotheses tested in this research was that sentences which are not directly related to the phone and phone feature can introduce noise. To test this hypothesis an additional experiment was carried out where only sentences which contained mentions of the mobile phone and a phone feature linked to it were processed to produce rankings of the phone features. Therefore, selection of the relevant sentences was based on the results of coreference resolution (Chapter 4) and relation extraction (Chapter 5). At first we ran coreference resolution to find all mentions of the concrete phone in the text. At the next stage we automatically annotated all phone characteristics in the text (Section 3.2). The machine-learning approach described in Chapter 5 provided information about whether a phone feature is linked to a phone mentioned in the same sentence or not. The new corpus comprised of 10,069 sentences in total: 1,938 sentences for the "no" category and 8,131 sentences for the "yes" category. From here on, this corpus will be referred to as "selected sentences".

6.4.2 Gold standard

For evaluation purposes we conducted a separate data collection which was aimed at constructing the gold standard. We wanted to use people's input to rank the features they find most important when choosing a phone. For this purpose, we developed a special drag-and-drop interface which allowed the users to choose the most important phone features. No special guidelines were given to participants except that they needed to pick the five most important features for them from a

given list. The features were displayed in a random order.

In order to prepare the initial list for ranking, we manually checked the features collected from Wikipedia infoboxes and removed those that were difficult to understand without further explanation. We decided to use features from Wikipedia infoboxes rather than lists collected using corpus-based approaches (described in 3.2.1), because features and values acquired from infoboxes were already used in our research for the automatic pre-annotation of the features in the corpus (Section 5.3.1).

We also had to limit the number of options we showed to a user, so that the interface stayed user-friendly and easy to use. For this reason, we decided to keep only those features that we felt to be most important, of which there were 47 in total. We collected a total of 170 answers and used this information to get a weighted ranked list of features by assigning to each feature a score that was equal to the number of times a feature was selected. Table 6.1 shows the top 20 features together with their frequencies.

6.4.3 Baseline

In order to evaluate the effectiveness of our ranking methods, we implemented two baselines. The first baseline considers the information from the infoboxes and ranks a phone feature on the basis of how many Wikipedia articles about mobile phones mention the feature in the infobox and assign it a value. The second baseline ranks a feature on the basis of how many times it is encountered in the Wikipedia articles describing mobile phones. In both cases, only exact matching

90	price
81	battery
57	operating system
52	phone style
42	manufacturer
31	size
29	standby time
29	GPS
25	connectivity
24	3g network speed
23	memory
22	network data connectivity
21	camera
21	talk time
20	weight
19	keyboard
19	main screen
19	touchpad
18	CPU
18	hardware platform

Table 6.1: Gold standard - the top 20 features together with their frequencies

is used when looking for features. By using these baselines, we can see whether a corpus of reviews is beneficial to us.

6.4.4 Evaluation metrics

Our intrinsic evaluation was based on comparing several rankings to each other, so we had to consider some formal metrics which would give us an objective number. We decided to choose two metrics that are commonly used to measure the association between two measured quantities.

The first one is the Kendall rank correlation coefficient and is commonly referred to as Kendall's tau coefficient Abdi (2007). It depends on the number

of inversions of pairs of objects which would be needed to transform one rank order into the other Abdi (2007). Equation 6.1 describes the formula used for calculating Kendall rank correlation coefficient.

$$\tau = \frac{N_c - N_d}{\frac{1}{2} * n * (n-1)} \tag{6.1}$$

where N_c is the number of concordant pairs, N_d is the number of discordant pairs, and n is the total number of pairs. τ takes values between -1 and 1, where -1 means that two rankings are the reverse of each other and 1 shows that rankings are the same.

The second metric we used is Spearman's rank correlation coefficient or Spearman's rho, which is a non-parametric measure of statistical dependence between two variables Maritz (1984). Spearman's rank takes into account differences between the ranks of each observation on the two variables and Equation 6.2 shows the way this metric can be calculated.

$$\rho = \frac{\sum_i (x_i - \overline{x}) * (y_i - \overline{y})}{\sqrt{\sum_i (x_i - \overline{x})^2 * (y_i - \overline{y})^2}} \tag{6.2}$$

Similar to Kendall's tau, the Spearman's rho values range from -1 to +1, and the closer to +1 they are, the more similar the rankings are.

The use of these metrics allowed us to output a score after comparing two lists. However, after several experiments we observed that these two metrics provide the same rankings; therefore, the results provided in the next section feature only Spearman's scores.

6.4.5 Results

As described in the previous sections, we carried out several experiments to produce different rankings of the features. We compared our rankings to the gold standard and the results of this comparison can be found in Table 6.2 and Table 6.3. For the evaluation, we used both the full gold standard and only the first 20 items in the gold standard. The justification for the second list is that it is highly unlikely that a customer will be willing to be asked about more than 20 features before they reach a decision.[1] We also present results of filtered full list and Top 20, in which we noticed that some features and their values are very difficult to map automatically in the texts, like features "price" and "3g network speed". Some similar features were used in the ranking and it was decided to merge them and sum up their scores; for example, the feature "screen" was combined with "main screen", "exterior screen" and "external display".

The rows $Baseline_1$ and $Baseline_2$ correspond to the two baselines introduced in Section 6.4.3. As can be seen, the results obtained with the first baseline are among the lowest, indicating that using frequency from infoboxes is not useful. However, it should be noted that the second baseline, featuring frequency in Wikipedia articles, is quite a good one. The three rows with labels starting with *Frequency from reviews* contain the results obtained using just frequency of features in the reviews, but employing different feature matching methods. The remainder of the rows contain the results of the methods that use the opinion classifier and different

[1] In reality, we hope that by using the ranking methods presented in this thesis, the number of steps needed in an interactive question answering system in order to find a phone will be much lower.

feature matching methods.

The following section will discuss results in more detail and will provide insight in error analysis.

Method	**Full list**	**Filtered Full**	**Top 20**	**Filtered Top 20**
Baseline_1	-0.155	-0.025	0.019	0.102
Baseline_2	0.010	0.129	0.177	0.367
Frequency from reviews exact match	0.327	0.311	0.252	-0.026
Frequency from reviews fuzzy match	0.164	0.215	0.293	0.090
Frequency from reviews values match	-0.014	0.159	0.172	0.243
Weighted frequency with exact match	0.326	0.231	0.357	0.138
Weighted frequency with fuzzy match	0.246	0.213	0.317	0.103
Weighted frequency with values match	0.006	0.181	0.263	0.382
Frequency from opinionated sentences exact match	0.235	0.180	0.211	0.119
Frequency from opinionated sentences fuzzy match	0.299	0.334	0.332	0.312
Frequency from opinionated sentences values match	-0.040	0.063	0.110	0.331
Frequency from negative sentences with exact match	-0.079	0.012	0.009	-0.004
Frequency from negative sentences with fuzzy match	-0.172	-0.158	-0.057	-0.162
Frequency from negative sentences with values match	-0.019	0.062	0.184	0.241
Frequency from positive sentences with exact match	0.019	0.024	0.130	-0.015
Frequency from positive sentences with fuzzy match	0.253	0.257	0.223	0.277
Frequency from positive sentences with value match	0.023	0.082	0.301	**0.532**

Table 6.2: The evaluation results for the full corpus

Method	**Full list**	**Filtered Full**	**Top 20**	**Filtered Top 20**
Baseline_1	-0.155	-0.131	0.019	0.102
Baseline_2	0.010	-0.025	0.177	0.367
Frequency from reviews exact match	0.293	0.346	0.367	0.469
Frequency from reviews fuzzy match	-0.088	-0.056	0.276	0.202
Frequency from reviews values match	-0.095	0.080	0.048	0.336
Weighted frequency with exact match	0.314	0.372	0.384	0.465
Weighted frequency with fuzzy match	-0.055	-0.023	0.294	0.211
Weighted frequency with values match	-0.103	0.078	0.046	0.336
Frequency from opinionated sentences exact match	0.300	0.365	0.388	0.466
Frequency from opinionated sentences fuzzy match	-0.057	-0.022	0.294	0.211
Frequency from opinionated sentences values match	-0.111	0.058	0.026	0.304
Frequency from negative sentences with exact match	0.044	0.053	0.309	0.244
Frequency from negative sentences with fuzzy match	-0.130	-0.135	0.077	-0.023
Frequency from negative sentences with values match	-0.131	-0.123	0.070	0.055
Frequency from positive sentences with exact match	0.124	0.251	0.063	0.191
Frequency from positive sentences with fuzzy match	0.008	0.027	0.314	0.296
Frequency from positive sentences with value match	-0.252	-0.130	-0.055	0.237

Table 6.3: The evaluation results for selected sentences

Rank	Method	Sum of ranks
1.	Weighted frequency with exact match	22
2.	Frequency from reviews exact match	35
3.	Frequency from opinionated sentences exact match	40
4.	Frequency from opinionated sentences fuzzy match	51
5.	Baseline_2	52
6.	Frequency from positive sentences with fuzzy match	52
7.	Weighted frequency with values match	62
8.	Weighted frequency with fuzzy match	65
9.	Frequency from reviews values match	70
10.	Frequency from reviews fuzzy match	72
11.	Frequency from positive sentences with value match	79
12.	Frequency from opinionated sentences values match	85
13.	Frequency from negative sentences with exact match	86
14.	Frequency from positive sentences with exact match	87
15.	Frequency from negative sentences with values match	104
16.	Baseline_1	114
17.	Frequency from positive sentences with fuzzy match	131

Table 6.4: Ranking of all methods

6.4.6 Discussion of results and error analysis

After examining the results of intrinsic evaluation we noticed that some methods perform much better than others in a range of settings. To find out which method outperforms others more objectively, we examined all the rankings and used their rank as a score. We summed up these scores for all different settings to get the final score describing how successful this ranking method was. Results are presented in Table 6.4, where the lowest score reveals the best method.

Table 6.4 reveals that weighted frequency with an exact match outperforms all other ranking methods. It is worth noting that the exact match setting seems to be the most effective, considering that the top three methods in this list are based on

exact matching. We believe it can be explained by the fact that it eliminates noise at the matching stage. Also the second and the third best performing methods are frequency from all sentences and frequency from opinionated sentences, which explains why combining these two methods and capturing it in weighted frequency gives the best result.

Tables 6.2 and 6.3 show that the evaluation list proved to be very important as well; precision of the methods improves when only the top 20 items are considered. This result can be explained by the fact that the top items get the highest frequency and, therefore, more data is available for them and their ranking is more stable, whereas the "long tail" is less predictable. Filtering, removing features that are very difficult to match in the texts, gives a fairer comparison of the gold standard and the results acquired. In the filtered list, the second baseline, based on matching features in Wikipedia articles, improves considerably (from 0.010 to 0.367), which indicates that filtering the gold standard provides a fairer comparison.

Whilst using selected sentences does not lead to better results, the results do become more stable in general. It shows stable results even on the whole list of features from the gold standard as opposed to using the whole corpus, which shows much worse results for the full list. Therefore, our assumption that it helps to remove noise seems valid and proven. It means that by removing sentences that do not contain mention of the phone and a feature linked to it, we can ensure we get better results.

6.4.6.1 Analysis of matching algorithms

The experiments carried out in this thesis revealed several problems to be tackled in order to obtain better results. One of the first issues we had to address when implementing the matching algorithm was the possibility to refer to the same feature in several different ways. For example, the feature *operating system* can be referred to using "Operating System", "operatingsystem" or "os". Even though we used WordNet and manually compiled lists, it is unlikely that we managed to cover all the possible ways people refer to a feature. For this reason, the fuzzy matching method is not always very precise. Related to this problem is the fact that the list of values of a feature is likely to grow over time. Unless these values are listed in Wikipedia and our matching algorithm gets updated there is no way to capture the mention of a corresponding feature in a review.

Another problem concerns the ambiguity of the features. For example, the features "standby time" and "usage time" have very similar meaning. It can be explained by the nature of Wikipedia resource which is built by different collaborators. Even though they are encouraged to use infobox templates, there is nothing to impose them use a standard way to refer to the features or values. This situation becomes even more problematic when the features are considered out of the context, as in the case of the experiment carried out to produce the gold standard. In light of this, word sense disambiguation-like methods could be considered to find out whether two similar expressions refer to the same feature on the basis of their context.

Another problem related to matching of features is connected to such pairs as "camera" and "video camera". When using only strict matching, it is difficult to decide whether the users just described a photo camera or whether they are referring to a photo-video camera. This problem becomes even more complicated when both forms are used in the text and "camera" is coreferential with "video camera". The only way to address this problem is to employ a coreference resolver.

The use of WordNet to obtain synonyms introduced a fair number of errors as well. For example, for the feature "carrier" some of the synonyms are "postman", "carrier wave", "mailman" and "attack aircraft carrier" which are completely unrelated to the features of mobile phones. This is due to the fact that the word used to refer to this feature is far too general and therefore ambiguous. At the other extreme are the features such as "hardware platform" which are too specific and do not appear in WordNet. For this reason, it will be necessary to produce a better list of synonyms for the features.

We should also acknowledge that we are fully aware of the fact that the error accumulation in our pipeline have affected the results as well and reduced the overall performance of the method. For example, while creating the corpus of selected sentences, errors were accumulated from coreference resolution, feature annotation and use of classifier. Even though at all stages we tried to achieve the highest precision possible, previous chapters of this thesis revealed that mistakes in the work of algorithms were inevitable due to the range of various factors.

6.5 Extrinsic Evaluation

6.5.1 Motivation

The intrinsic evaluation carried in Section 6.4 showed that our method outperforms two baselines. However, the evaluation was based on the assumption that the gold standard ranking produced by people would be the most effective when used for IQA. The gold standard was based on users' perceptions of the relevance of the features, though it was unclear whether it enabled the phone to be found quickly. Keeping in mind that our initial motivation was to use our ranking methods for creating IQA sessions, we decided to carry out an extrinsic evaluation of our method as well.

As described in Chapter 1, our research sought to find the best way to navigate users in the search for a product. Our hypothesis was that the use of feature ranking can contribute to the speed of taking a decision and make the search for information easier and quicker. Therefore, we looked for objective ways to measure whether the use of rankings produced by our methods makes IQA more efficient. The following section will describe the experiment carried out in more detail.

6.5.2 Experiment

As mentioned in Chapter 1, we were not intending to build the whole IQA system and were rather focusing solely on the dialogue management component. Therefore, we decided to approximate other components of the system and only focus on the search for information in terms of IQA. We were interested to see whether rankings contribute to the speed the decision is taken when searching for

a product. It can be done in several ways, for example, as described in Section 2.7, we could look at how quickly an answer to our request can be found or how many iterations in terms of IQA we may need to get the correct answer (Jurafsky and Martin, 2009; Harabagiu et al., 2005).

Since we did not focus on having a fully-fledged system to present to people, it was decided to focus on automatic evaluation of the decision taking. For this purpose, we had compiled a database of mobile phones based on the Wikipedia corpus of infoboxes (mentioned in Section 3.1.1). Each infobox featured the name of the phone and characteristics assigned to it (features with corresponding values). In this way we were able to assemble a database describing 450 mobile phones.

At the next step, the algorithm would randomly select a phone from our database and attempt to find it based on different features. At each step the algorithm takes one feature of the phone as a search criterion and in this way limits the search space. It uses the feature value assigned to the randomly chosen phone we are looking for and selects all the phones with the same value. For example, the randomly chosen phone may have a feature "OS" ("operating system") with a value "Android" assigned to it; therefore at some point we will pick feature "OS" and will look for all phones that have Android OS. As we add more features, the search space becomes smaller and at the final step (in case of the successful search) we should obtain the single phone we were looking for. However, at the point of picking a feature to be searched for, we can use a different order; specifically, we can employ rankings generated by our methods to decide which features should be searched for first to ensure efficient decision taking in terms of IQA.

For our experiment we have limited the list of features in our database to those that appeared in the gold standard described in Section 6.4 and were later employed for ranking. In addition, only those features that had a range of different values present in infoboxes were selected. We made sure to map all the features if they were expressed in several ways in the initial corpus of infoboxes, for example, "operating_system", "os" and "operatingsystem" were merged with feature "operating system". Values of these feature were combined as well. The final list of features comprised of 21 items.

We also performed mapping of the values. Infoboxes are created by different people and therefore the ways the same values can be expressed in the infobox can differ considerably. At the initial stage, we employed the same heuristics as described in Section 3.2 to get a full list of various values assigned to a feature. At the next step, we attempted to generalise values in order to get a shorter and more consistent list. For example, instead of adding value "Android OS 1.5" or "Symbian OS 9.1" to our database, we mapped them to more general categories - "Android" and "Symbian" accordingly.

The compiled database was used to evaluate how quickly a decision can be taken using different ways to choose features to search for. The evaluation results are described in the next section.

6.5.3 Evaluation results

In order to evaluate how effective the use of rankings was, we implemented a baseline that was based on choosing features from the initial list in a random way.

It was also decided to test how efficient the search would be if we employed the gold standard ranking described in the previous section. By using these methods, we can have a reference point and compare more objectively how employing our rankings affects the decision making.

Our experiment was based on calculating the number of steps it takes to find a phone that we had initially chosen. As previously mentioned, we selected the phone in a random way and this experiment was repeated 2000 times for each ranking method. Sometimes this search was not successful and then the results of this attempt were ignored. We calculated the number of turns only for the successful cases, when the phone was found by narrowing the search space. Then the average performance in these 2000 experiments was taken. We also calculated the standard deviation in order to see that the performance of the methods employed is stable and reliable.

The results of these experiments are presented in the Table 6.5, where "full corpus" refers to the rankings obtained using the full corpus of user reviews, whereas "selected" are based on rankings obtained using the corpus of selected sentences featuring both a mobile phone and a feature linked to it. We also used "filtered" versions of the rankings that were described in the previous section. These results are discussed and analysed in more detail in the following section.

Method	**Full list**	**Filtered Full**	**Selected**	**Filtered Selected**
Random selection	13.26/4.657			
Gold standard	10.89/4.50	11.36/4.49	10.99/2.81	11.21/4.56
Baseline_1	11.34/2.93	11.28/2.89	10.96/2.80	11.29/3.03
Baseline_2	14.16/3.83	14.14/3.98	14.11/4.16	14.14/3.91
Frequency from reviews exact match	13.46/4.43	13.11/4.47	13.56/5.07	13.14/4.60
Frequency from reviews fuzzy match	11.02/4.49	11.02/4.52	11.44/4.71	10.98/4.43
Frequency from reviews values match	12.65/3.95	13.07/3.88	13.40/4.65	12.90/3.95
Weighted frequency with exact match	13.17/4.65	13.31/4.53	13.22/5.05	13.31/4.55
Weighted frequency with fuzzy match	10.82/4.43	10.72/4.49	11.42/4.83	11.01/4.56
Weighted frequency with values match	12.99/3.89	13.08/3.89	13.43/4.61	13.05/3.94
Frequency from opinionated sentences exact match	13.04/4.73	13.36/4.61	13.18/5.07	13.18/4.63
Frequency from opinionated sentences fuzzy match	10.63/4.30	10.78/4.44	10.98/4.53	10.93/4.51
Frequency from opinionated sentences values match	12.98/3.88	12.88/3.86	13.36/4.59	12.96/3.88
Frequency from negative sentences with exact match	12.38/4.52	12.01/4.60	12.59/5.08	12.49/4.52
Frequency from negative sentences with fuzzy match	10.51/4.33	10.72/4.56	11.08/4.58	10.59/4.56
Frequency from negative sentences with values match	12.91/3.95	12.99/4.01	13.48/4.73	12.71/4.01
Frequency from positive sentences with exact match	11.02/4.36	13.40/4.90	13.08/5.23	13.37/4.72
Frequency from positive sentences with fuzzy match	12.88/4.01	11.19/4.41	11.19/4.76	11.19/4.57
Frequency from positive sentences with value match	14.16/3.83	13.20/3.85	13.42/4.64	13.06/3.82

Table 6.5: Results for extrinsic evaluation

6.5.4 Discussion

The extrinsic evaluation revealed that our rankings in some cases not only outperform the baseline, but also perform slightly better than the gold standard generated by humans. Therefore, even though the gold standard represents users' perceptions, it can be concluded that using completely automatic methods of feature ranking yields better results. This gives indications as to why there is no correlation with an intrinsic evaluation described earlier in this chapter. If we consider the gold standard as not the most efficient ranking, then the results of intrinsic evaluation can be challenged. We also regard extrinsic evaluation as more objective in this case. For example, unlike intrinsic evaluation which favoured exact match, the extrinsic evaluation reveals that the fuzzy matching is the most efficient when producing rankings to be employed in IQA. However, the extrinsic evaluation did not reveal any major difference between the performance of rankings produced using the whole corpus and the selected sentences. Filtered versions of the rankings in the majority of the cases yield better results. The use of opinionated sentences improves the overall performance of the search for a product.

It should also be mentioned that, for all the rankings, standard deviation is quite low, which shows that the algorithm's performance is stable and there is no huge variation in the number of steps it takes to take a decision.

We understand that the evaluation used in this section is only an approximation of the real IQA where the user also has an initiative; nonetheless, it shows that we can enhance the user's experience without investing much manual labour to

gather real interactions in terms of IQA sessions. Therefore, we consider that our method can be used to improve the decision making in IQA systems oriented at product search.

6.6 Conclusions

This chapter addressed the problem of feature ranking for interactive question answering systems which help customers to choose the right product for them. In terms of intrinsic and extrinsic evaluation, two baselines and several ranking methods were evaluated against a gold standard collected from users. An experiment showed that the automatic ranking methods proposed outperform the baseline and show promising results. The evaluation also confirmed the fact that using a corpus of reviews is beneficial for feature ranking and, therefore, for improving the efficiency of IQA. The results were further improved by using only the opinionated sentences for scoring features.

Chapter 7

Conclusions

The research presented in this thesis focused on the field of interactive question answering and explored different ways in which the search for products can be enhanced using NLP techniques. This chapter summarises the original contributions of this research and how the goals set up in Chapter 1 were addressed.

Section 7.1 revisits the goals of this thesis to discuss how they were achieved in our research. It is followed by Section 7.2 which presents the main contributions of this thesis. Section 7.3 summarises the content of the thesis chapter by chapter, and directions for future work are discussed in Section 7.4.

7.1 Hypotheses and goals revisited

This section examines the goals described in Chapter 1 and discusses the way they were achieved in this research. The main **aim** of this thesis was to identify ways to design an interactive question answering system that will assist users in choosing a product in an optimal way. In order to fulfill this aim, we tested several research hypotheses and accomplished the goals set up in Chapter 1.

We tested several research hypotheses in this thesis:

- **Hypothesis 1**: It is possible to use a data-driven approach for designing an IQA system.

  Chapters 3 and 6 presented the use of corpus-based methods for identifying features in the text and creating their rankings to be used for an IQA system. This hypothesis was proven; however, the limitations of the data-based approach were discussed as well.

- **Hypothesis 2**: Product characteristics mentioned in user reviews are important for a person who is likely to purchase a product and can therefore be used when designing an IQA system. A ranking of product features can be used to provide the IQA system with information on which product characteristics should be given priority and presented first.

  Chapter 6 described several experiments that were carried out to generate rankings of product features. The extrinsic evaluation revealed that using the produced rankings can reduce the time it takes to make decisions and search for products. During the extrinsic evaluation our methods outperformed the baseline. Extrinsic evaluation also revealed the difficulties of intrinsic evaluation that relies on the gold standard representing users' perceptions. If we consider the gold standard as not the most efficient ranking, then the results of intrinsic evaluation can be challenged.

- **Hypothesis 3**: Only sentences which are directly related to the phone and phone feature should be used when ranking the features in order to reduce noise.

  Chapter 4 helped to find all mentions of the phone in the texts, whereas

Chapter 5 identified whether there was a link between these mentions and the phone features. This information was used to select only sentences related to the phone and the phone features linked to it. This hypothesis was tested in Chapter 6 by producing rankings using only the selected sentences. Intrinsic evaluation revealed that whilst using selected sentences does not lead to better results, the results do become more stable in general. Therefore, our hypothesis that it helps to remove noise seems valid and proven. However, extrinsic evaluation did not reveal any major difference between the performance of rankings produced using the whole corpus and the selected sentences.

- **Hypothesis 4**: Opinionated sentences are a good source when it comes to ranking the phone features.

  This hypothesis was tested in Chapter 6 by using information about sentence polarity when producing feature rankings. Intrinsic evaluation proved that the use of opinionated sentences is beneficial for our methods.

In order to achieve the main aim of this thesis and test the hypotheses, several **goals** had to be achieved. We revisit these goals and describe how they were accomplished.

- **Goal 1** was to investigate the field of IQA and review the current approaches to the design of IQA systems.

  This goal was achieved by carrying out a detailed review of the state of the art in IQA field, which is described in Chapter 2.

- **Goal 2** was to collect and annotate resources to be used for this research.

  Several chapters contributed to addressing this goal. Chapter 3 described the collection of two corpora: Wikipedia articles and customer reviews describing mobile phones. These corpora were annotated with corefential links and this process is discussed in Chapter 4. The review corpus was also annotated for links between mentions of the phone and its features. This annotation is presented in Chapter 5.

- **Goal 3** was to develop a coreference resolution module that identifies all mentions of the product in the text.

  This goal was achieved in Chapter 4, which described coreference resolution methods. The coreference resolution was developed first for Wikipedia articles, and then this approach was adapted to the review corpus.

- **Goal 4** was to develop a machine learning-based relation extractor, which is meant to be used to find links between mentions of the phone and phone features.

  This goal was achieved by developing a machine-learning method to identify links, which is discussed in Chapter 5.

- **Goal 5** was to investigate whether product characteristics mentioned in user reviews are important for a person who is likely to purchase a product and can therefore be used when designing an IQA system.

  This goal was accomplished in Chapter 6, where we explored different methods to rank product features in order to be able to provide the IQA system with information about which product characteristics should be given

priority and presented first. These rankings are presented and evaluated in Chapter 6.

7.2 Original contributions

The achievement of the goals described in the previous section allowed us to make several original contributions to the field of natural language engineering and IQA, in particular.

The first original contribution is the development of a new approach to enhance the performance of IQA. The literature review in the field of IQA revealed that there is a need to enhance the performance of current IQA systems. We focused on the product domain, which, we believe, will greatly benefit from using IQA systems. The suggested strategy of ranking features in terms of IQA systems targeted at product search is a novel one, and, to the best of our knowledge, was first suggested in this thesis. This strategy is based on the hypothesis that the ranking of features, in order to identify which ones to present first to the user, can make IQA systems more efficient. This approach was tested by exploring different ways to produce rankings. Then, these rankings were evaluated against the gold standard compiled with the help of human contributors. Another evaluation was carried out to find out whether the suggested approach makes the search for products more efficient. It tried to replicate the way humans can search for a product using an IQA and how their search can be assisted using the ranking of product features. We believe that this evaluation is closer to the reality and therefore should be more objective. The experiments revealed that the method

suggested in this thesis can be used instead of the rankings collected using human input. This can make the construction of IQA easier, less labour-intensive and time-consuming. The best ranking methods are based on the weighted frequency of the features in the review corpus, which takes into account both opinionated and neutral sentences. Furthermore, the use of the rankings improves the speed of taking a decision when choosing a mobile phone.

The second original contribution is the development of a rule-based coreference resolution, which links all mentions of the phone in the text. We developed a coreference resolution approach that was used for two types of texts: Wikipedia and review texts. The suggested method helps to identify corefential links describing the topic of the article, a mobile phone in our case. The coreference resolution method is based on a set of rules that take into account the particularities of the ways phones can be referred to in the texts. The evaluation showed that these method obtain f-measure, which is comparable to the state-of-the-art systems. The use of this approach for two different types of texts proved that the suggested method can be adapted for other corpora. The developed coreference resolution module can be also adapted to deal with other domains of products.

The third original contribution is the development and evaluation of a machine-learning method that identifies links between the mentions of the phones and their features. We explored different machine-learning features inspired by relation extraction literature in order to find the best combination. This method takes as input a sentence with mention of the phone and the phone features

annotated. Then, it makes a judgement about whether there is a link between the mention of the phone and the phone feature. The best results of the classification are achieved by employing the following set of ML features: the name of the phone feature, head of the name of the feature, indicating phrase, distance, distance (threshold = 9), possessive relation, prepositional relation and verb dependency (two levels up). This approach can be used to identify features that belong to the mobile phones mentioned in the texts.

The fourth original contribution consists of the resources compiled for the experiments carried out in this thesis:

- a corpus of Wikipedia articles describing mobile phones (texts and infoboxes), consisting of 560 articles (216,555 words);
- a review corpus describing mobile phones, comprising of 3,392 reviews (114,708 sentences; 2,253,877 words);
- a corpus of Wikipedia articles annotated with markables and coreferential chains for the main topic, 20 documents with almost 22,000 words. In our corpus we annotated a total of 668 coreferential relations, 83 "set of" relations and 59 "alias" relations;
- a corpus of review articles annotated with coreference for the main topic and features of the phone, 20 documents with almost 77,500 words. We annotated 1075 coreferential relations, 19 "set of" relations and only two "alias" relations. 4809 features were annotated in this corpus as well;
- a corpus of review texts annotated with links between mentions of the phone

and features. This corpus comprises of 40 files and 984 links annotated;

- a database of phones and their features compiled using Wikipedia infoboxes. This database describes 450 mobile phones using a set of features and their corresponding values.

An additional contribution of this thesis is a literature review in the field of interactive question answering. This field does not have a lot of review articles/book chapters describing the state of the art of the area. This is why such a description of the relevant work is a valuable original contribution to the NLP field on its own. This review discussed not only IQA systems, but also referred briefly to the related fields such as dialogue and QA systems. It allowed us to introduce the basic concepts that are common for these fields. This review also revealed that the definition of an IQA system is debatable and it is sometimes difficult to draw a clear distinction between IQA and dialogue systems. We argue that any IQA system can be considered a dialogue system, but only those dialogue systems which are information-seeking can be regarded as IQA systems. This review also covered different ways of developing IQA systems: by producing a scaled-down version of information-seeking dialogue or by integrating additional functionalities into a standard QA system. We also discussed another approach where the system interacts with users by proposing which questions it can answer next, therefore exploring the possibility to use follow-up questions. This literature review revealed that the field of IQA has a lot of research questions to be addressed, and there is much room for improvement for current systems.

7.3 Review of the thesis

This section provides an overview of this thesis by briefly summarising each chapter.

Chapter 1 introduced the topic of research for the thesis. It also provided details about the initial motivation of this investigation and its importance for the real world. This chapter discussed the choice of the domain for our research, we focused our attention on the domain of products and, more specifically, on the subdomain of mobile phones. This chapter also featured an example of a dialogue that could take place between a shop assistant and a customer, followed by the discussion on how it could be accommodated by an IQA session. We set up goals to be achieved by the research carried out in this thesis and outlined the original contributions of this research.

Chapter 2 discussed the field of interactive question answering. It also provided background information about the fields of QA and dialogue systems, which are closely related to IQA. This chapter also outlined the constraint-based approach to IQA, which is adopted in this research. It highlighted a lack of research in this area and at the same time pointed out a big potential of the development of the investigation in this direction.

Chapter 3 discussed the data employed in this research, specifically, two types of corpora: semi-structured texts and unstructured texts. It also provided more details about the way these corpora were collected. These corpora were used for the research described in chapters 4, 5 and 6. This chapter also addressed the

problem of the extraction of features describing mobile phones. We explored two types of methods that can be used to tackle this task, and discovered that the corpus-based approach is not very helpful. Our corpus was not big enough to give good results with this methodology. As we aimed to use more automatic methods, we decided to employ semi-structured resources, specifically, Wikipedia infoboxes. This strategy allowed us to get a list of phone features and corresponding values, which were later used in Chapters 5 and 6.

Chapter 4 addressed the problem of coreference resolution for two types of corpora discussed in Chapter 3. It introduced the field of information extraction and presented the state of the art in the coreference resolution. This chapter described all stages of development of the coreference resolution, including the elaboration of the annotation guidelines and the annotation of the corpora for coreference phenomena. It also provided insight into the motivation to develop our own coreference resolution system rather than use already existing systems. The results of our approach were promising for most of the texts as it relied on the presence of a regular structure in the articles as well as special language used to talk about the product in question. The research carried out in this chapter also proved that it is possible to use the proposed method for different kinds of texts.

Chapter 5 described the identification of links between mentions of the phone and its features. It provided an overview of relation extraction and machine learning. This chapter also described the corpus annotation used for this task. A detailed description of the ML features which were used for the relation extraction, and the way they were extracted, is provided as well. The chapter presented a

ML method and its evaluation carried out using 5-fold cross-validation. The best baseline achieved 61.08% accuracy, whereas the best combination of the features for our machine learning achieved 73.28% accuracy. The output of the classifier developed is used in Chapter 6 for the extraction of sentences containing mentions of the phone and phone characteristics linked to them.

Chapter 6 presented the feature ranking for interactive question answering systems that can assist users in choosing a product. It presented relevant work and highlighted the novelty of the approach adopted by our research. This chapter also focused on the description of the various ranking methods developed and their evaluation. We carried out two types of evaluation: intrinsic and extrinsic. Intrinsic evaluation compared our rankings to the gold standard collected using human input. It showed that the methods suggested in our research obtain better results than both baselines. However, we were not sure how reliable our gold standard was; therefore, an extrinsic evaluation was carried out as well. We compared the performance of product search when different rankings were used to choose phone features. The results revealed that our rankings in some cases not only outperform the baseline, but also perform slightly better than the gold standard generated by humans. Therefore, our method can be used to enhance IQA systems orientated at product search.

7.4 Future work

The research described in this thesis revealed several possible directions for future work. They are discussed in more detail in this section.

In Chapter 3 we discussed the need to find better ways to extract product features. The first way to approach this problem will be to identify methods for more efficient cleaning of Wikipedia infoboxes from the noise. For example, one of the problems we encountered was connected to the lack of consistency in referring to the features and values in the infoboxes. To address this issue, we can attempt merging similar product features based on the similarity of their values. It can be done in a semi-automatic way: we can at first cluster features based on their values and then check manually whether we really want to merge some of them. Word sense disambiguation-like methods could be also considered to find out whether two similar expressions refer to the same feature on the basis of their context. Another direction of the research could be to try other methods to identify features, e.g., framework similar to (Bhattarai et al., 2012) can be employed. It can also benefit from the review corpus we have collected, and this corpus can be enlarged with relative ease.

The automatic annotation of the phone features described in Chapter 3 revealed that coreference resolution for all markables in the texts is needed to make the annotation more precise. For example, coreference resolution will help to identify whether the annotated feature "camera" refers to the previously mentioned "video camera" or "photo camera".

Another direction of our research would be to try enlarging and updating our database of phones using a bigger corpus of texts. We can identify a mention of the phone in the text and also whether the feature or the value mentioned in the same sentence belongs to it. It would allow us to discover new values/features

that are not yet added to the database. The same approach can be used to check how reliable our database is: if the phone mention is never encountered with this feature/value in the texts, maybe it is an indication that this information is not up-to-date.

At this stage of our research, we were not able to benefit from the Wikipedia markup. It would be interesting to investigate this direction, and explore, for example, whether some feature values that contain links to other Wikipedia pages are more reliable that the ones without links. We can also use these links to merge similar values – if two values contain links which point to the same page, it can be an indication that they are identical or similar. For example, we can merge "[[QWERTY]] keypad" and "[[QWERTY]] keyboard/numerical" because they point to the same page "QWERTY".

Another area not explored in this research is the information that we have for each review in our review corpus, i.e. the user name of the reviewer, the time the review was submitted, the name and the brand of the product reviewed, as well as the star rating. This additional information is not currently used in our research but may be interesting for additional investigations in the future.

Chapter 4 discussed the possibility of using the developed coreference resolution method for other kind of texts. One interesting line of research would be testing our coreference resolution method for other kinds of products and to see how easy it is port it to another domain.

Another line of research is to try relation extraction on the Wikipedia corpus. We already have a coreference resolution module for Wikipedia texts. The

same automatic method of feature annotation, which was used for the review corpus, can be employed. It would provide the required input for our classifier. The investigation of this attempt may provide valuable insight into whether our approach is applicable to different kinds of texts or not.

We are also interested to explore new ways of further evaluating the suggested approach of using feature rankings for IQA systems. The next step of evaluation could be to investigate how using our rankings contributes to the naturalness of the user-system interaction. However, in order to carry out these experiments, fully-fledged IQA should be developed first and evaluated by the users using, for example, questionnaires as described in Chapter 2. This direction of the research seems very promising, and, in the long run, can lead to the development of a fully-fledged IQA system, which will assist users in choosing products. The authors hope that such a system will help customers to save time and make better choices when selecting a product to buy.

Appendix A

Previously published work

Some of the work described in this thesis has been previously published in the proceedings of peer-reviewed international conferences or books. However, before the inclusion in this thesis, the work has been extended or modified to account for new research directions pursued and to adapt to the context of this thesis. This appendix provides the list of the previously published work as well as a brief explanation of their contribution to this thesis:

- Konstantinova, N. and Orasan, C. (2011) Issues in topic tracking in Wikipedia articles. In Proceedings of the International Conference on Knowledge Engineering, Principles and Techniques (KEPT2011), Cluj-Napoca, Romania, July 46

  This paper presents the corefence resolution method developed for Wikipedia articles. It is described in more detail in Chapter 4. In this thesis a new evaluation metric is also used to evaluate its performance.

- N. Konstantinova, C. Orasan and P. P. Balage (2012) A Corpus-Based Method for Product Feature Ranking for Interactive Question Answering Systems. In International Journal of Computational Linguistics and Applications. Volume 3, Issue 1, pp. 57 - 70

This paper present the initial steps of developing ranking methods described in Chapter 6. This research was considerably extended and modified to incorporate different types of corpora and also the extrinsic evaluation.

- Konstantinova, N. and Orasan, C. (2013) "Interactive Question Answering." In S. Bandyopadhyay, S. Naskar, & A. Ekbal (Eds.), Emerging Applications of Natural Language Processing: Concepts and New Research, IGI Global, pp. 149-169, doi:10.4018/978-1-4666-2169-5.ch007

  This book chapter is a preliminary version of the Chapter 2. However, the chapter of this thesis was updated to include more recent advances in the field and discuss in more detail the relevance of this research for this thesis.

Bibliography

Hervé Abdi. Kendall rank correlation. In N.J. Salkind, editor, *Encyclopedia of Measurement and Statistics*, pages 508–510. Thousand Oaks (CA): Sage, 2007.

Peggy M. Andersen, Philip J. Hayes, Alison K. Huettner, Linda M. Schmandt, Irene B. Nirenburg, and Steven P. Weinstein. Automatic extraction of facts from press releases to generate news stories. In *Proceedings of the Third Conference on Applied Natural Language Processing*, pages 170–177, 1992.

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, and Zachary Ives. DBpedia: A nucleus for a web of open data. In *Proccedings of 6th Intl Semantic Web Conference*, pages 11–15, Busan, Korea, 2007. Springer.

Nikhil Bansal, Avrim Blum, and Shuchi Chawla. Correlation clustering. In *Proceedings of the 43rd Symposium on Foundations of Computer Science*, FOCS '02, page 238, Washington, DC, USA, 2002. IEEE Computer Society.

Raffaella Bernardi and Manuel Kirschner. Context Modeling for IQA: The Role of Tasks and Entities. In *Proceedings of Workshop for Knowledge and Reasoning for Answering Questions (KRAQ08)*, Manchester, UK, 2008.

Raffaella Bernardi, Manuel Kirschner, and Zorana Ratkovic. Context fusion: The role of discourse structure and centering theory. In *Proceedings of the Seventh*

conference on International Language Resources and Evaluation (LREC10), Valletta, Malta, 2010.

Archana Bhattarai, Nobal Niraula, Vasile Rus, and King-Ip Lin. A domain independent framework to extract and aggregate analogous features in reviews. In *Proceedings of CICLING 2012*, volume LNCS 7181, pages 568–579. Springer, 2012.

Matthew W. Bilotti and Eric Nyberg. Evaluation for scenario question answering systems. In *Proceedings of the International Conference on Language Resources and Evaluation*, 2006.

Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python*. O'Reilly Media, 2009.

Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

Sasha Blair-Goldensohn, Kathleen McKeown, and Andrew Hazen Schlaikjer. Answering definitional questions: A hybrid approach. In *New Directions in Question Answering*, pages 47–58, 2004.

Lou Boves and Els den Os. Interactivity and multimodality in the IMIX demonstrator. In *ICME*, pages 1578–1581, 2005.

Sergey Brin. Extracting patterns and relations from the world wide web. In

Proceedings of the First International Workshop on the Web and Databases, pages 172–183, March 1998.

Trung H. Bui. Multimodal dialogue management - state of the art. Technical report, Enschede, Centre for Telematics and Information Technology, University of Twente, Enschede, 2006.

Razvan Bunescu and Raymond Mooney. Subsequence kernels for relation extraction. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 171–178. MIT Press, Cambridge, MA, 2006.

Junkuo Cao and Xuanjing Huang. Answering definitional question by dependency-based knowledge. In *Natural Language Processing and Knowledge Engineering, 2008. NLP-KE '08.*, pages 1–6, 2008.

Jean Carletta. Assessing agreement on classification tasks: the kappa statistic. *Computational Linguistics*, 22(2):249–254, June 1996.

Roberta Catizone, Andrea Setzer, and Yorick Wilks. State of the art in dialogue management. Deliverable, EU IST 5th Framework Project COMIC, 2002.

Yee Seng Chan and Dan Roth. Exploiting syntactico-semantic structures for relation extraction. In *Proc. of the Annual Meeting of the Association of Computational Linguistics (ACL)*, Portland, Oregon, 2011.

Charles L. A. Clarke, Gordon V. Cormack, D. I. E. Kisman, and Thomas R.

Lynam. Question answering by passage selection (multitext experiments for TREC-9). In *TREC*, 2000.

Jim Cowie and Wendy Lehnert. Information extraction. *Communications of the ACM*, 39(1):80–91, January 1996.

Aron Culotta, Andrew McCallum, and Jonathan Betz. Integrating probabilistic extraction models and data mining to discover relations and patterns in text. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 296–303, New York, New York, 2006. Association for Computational Linguistics.

Aron Culotta, Michael Wick, Robert Hall, and Andrew Mccallum. First-order probabilistic models for coreference resolution. In *Proceedings of HLT-NAACL 2007*, 2007.

Robert Dale, Hermann Moisi, and Harold Somers, editors. *Handbook of Natural Language Processing*. Marcel Dekker, Inc., 2000.

Pascal Denis and Jason Baldridge. A ranking approach to pronoun resolution. In *Proceedings of the 20th international joint conference on Artifical intelligence*, pages 1588–1593, San Francisco, CA, USA, 2007. Morgan Kaufmann Publishers Inc.

Iustin Dornescu and Constantin Orăsan. Interactive QA using the QALL-

ME framework. *International Journal of Computational Linguistics and Applications*, 1(1–2):233 – 247, 2010.

Micha Elsner and Eugene Charniak. The same-head heuristic for coreference. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 33–37, Uppsala, Sweden, July 2010. Association for Computational Linguistics.

Oren Etzioni, Michael Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. Weld, and Alexander Yates. Unsupervised named-entity extraction from the web: an experimental study. In *Artificial Intelligence*, volume 165, pages 91–134, Essex, UK, 2005. Elsevier Science Publishers Ltd.

Oren Etzioni, Michele Banko, Stephen Soderland, and Daniel S. Weld. Open information extraction from the web. *Communications of the ACM*, 51:68–74, December 2008.

Christiane Fellbaum, editor. *WordNet: An Electronic Lexical Database*. The MIT Press, 1998.

Thomas Finley and Thorsten Joachims. Supervised clustering with support vector machines. In *Proceedings of the 22nd international conference on Machine learning*, ICML '05, pages 217–224, New York, NY, USA, 2005. ACM.

Pamela Forner, Anselmo Peñas, Eneko Agirre, Iñaki Alegria, Corina Forascu, Nicolas Moreau, Petya Osenova, Prokopis Prokopidis, Paulo Rocha, Bogdan Sacaleanu, Richard F. E. Sutcliffe, and Erik Tjong Kim Sang. Overview of the CLEF 2008 multilingual question answering track. In Carol Peters, Thomas

Deselaers, Nicola Ferro, Julio Gonzalo, Gareth J. F. Jones, Mikko Kurimo, Thomas Mandl, Anselmo Peñas, and Vivien Petras, editors, *CLEF*, volume 5706 of *Lecture Notes in Computer Science*, pages 262–295. Springer, 2008.

J. Fukumoto, F. Masui, M. Shimohata, and M. Sasaki. Oki electric industry: Description of the Oki system as used for MUC-7. In *Proceedings of the 7th Message Understanding Conference*, 1998.

Olivier Galibert, Gabriel Illouz, and Sophie Rosset. Ritel: an open-domain, human-computer dialog system. In *INTERSPEECH*, pages 909–912, 2005.

Roberto Garigliano, Agnieszka Urbanowicz, and David J. Nettleton. University of Durham: Description of the LOLITA system as used in MUC-7. In *Proceedings of the 7th Message Understanding Conference*, 1998.

Arnaud Grappy and Brigitte Grau. Answer type validation in question answering systems. In *Proceedings of RIAO '10 Adaptivity, Personalization and Fusion of Heterogeneous Information*, RIAO '10, pages 9–15, Paris, France, France, 2010.

Ralph Grishman. Information extraction: techniques and challenges. In *In Information Extraction (International Summer School SCIE-97*, pages 10–27. Springer-Verlag, 1997.

Ralph Grishman and Beth Sundheim. Message Understanding Conference-6: a brief history. In *Proceedings of the 16th conference on Computational linguistics*, pages 466–471, Morristown, NJ, USA, 1996. Association for Computational Linguistics.

Sanda Harabagiu and Dan Moldovan. Question answering. In Ruslan Mitkov, editor, *Oxford Handbook of Computational Linguistics*, chapter 31, pages 560 – 582. Oxford University Press, 2003.

Sanda Harabagiu, Andrew Hickl, John Lehmann, and Dan Moldovan. Experiments with interactive question-answering. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 205–214, Ann Arbor, Michigan, 2005.

Laura Hasler, Constantin Orăsan, and Karin Naumann. NPs for Events: Experiments in Coreference Annotation. In *Proceedings of the 5th edition of the International Conference on Language Resources and Evaluation (LREC2006)*, pages 1167 – 1172, Genoa, Italy, May, 24 – 26 2006.

Marti A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th conference on Computational linguistics*, pages 539–545, Morristown, NJ, USA, 1992. Association for Computational Linguistics.

Grahame Hirst. *Anaphora in Natural Language Understanding.* Springer Verlag, Germany, 1981.

Eduard Hovy, Zornitsa Kozareva, and Ellen Riloff. Toward completeness in concept extraction and classification. In *EMNLP '09: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 948–957, Morristown, NJ, USA, 2009. Association for Computational Linguistics.

Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In

Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '04, pages 168–177, New York, NY, USA, 2004. ACM.

Kevin Humphreys, Robert Gaizauskas, Saliha Azzam, Christian Huyck, Brian Mitchell, Hamish Cunningham, and Yorick Wilks. University of Sheffield: Description of the LaSIE-II System as Used for MUC-7. In *Proceedings of MUC-7*, 1998.

Abraham Ittycheriah, Martin Franz, Wei jing Zhu, Adwait Ratnaparkhi, and Richard J. Mammone. IBM's statistical question answering system. In *Proceedings of the Tenth Text REtrieval Conference (TREC)*, 2000.

Zhu Junwu, Li Bin, Wang Fei, and Wang Sicheng. Mobile ontology. *JDCTA: International Journal of Digital Content Technology and its Applications*, 4(5): 46–54, 2010.

Daniel Jurafsky and James H. Martin. *Speech and language processing an introduction to natural language processing, computational linguistics, and speech recognition*. Prentice-Hall, Inc., 2nd edition edition, 2009.

Nanda Kambhatla. Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. In *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*, page 22, Morristown, NJ, USA, 2004. Association for Computational Linguistics.

Diane F. Kelly, Paul B. Kantor, Emile L. Morse, J. Scholtz, and Ying Sun.

Questionnaires for eliciting evaluation data from users of interactive question answering systems. *Natural Language Engineering*, 15(1):119–141, 2009.

Manuel Kirschner and Raffaella Bernardi. Exploring topic continuation follow-up questions using machine learning. In *Proceedings of NAACL HLT 2009: Student Research Workshop*, Boulder, CO, 2009.

Manuel Kirschner, Raffaella Bernardi, Marco Baroni, and Le Thanh Dinh. Analyzing Interactive QA Dialogues using Logistic Regression Models. In *Proceedings of XIth International Conference of the Italian Association for Artificial Intelligence (AI*IA09)*, Reggio Emilia, Italy, 2009.

Natalia Konstantinova and Constantin Orăsan. Interactive Question Answering. In Sivaji Bandyopadhyay, Sudip Kumar Naskar, and Asif Ekbal, editors, *Emerging Applications of Natural Language Processing: Concepts and New Research*, pages 149–169. IGI Global, Hershey, 2013.

Zornitsa Kozareva and Eduard Hovy. Learning arguments and supertypes of semantic relations using recursive patterns. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1482–1491, Uppsala, Sweden, July 2010. Association for Computational Linguistics.

Zornitsa Kozareva, Ellen Riloff, and Eduard Hovy. Semantic class learning from the web with hyponym pattern linkage graphs. In *Proceedings of ACL-08: HLT*, pages 1048–1056, Columbus, Ohio, June 2008. Association for Computational Linguistics.

James Lester, Karl Branting, and Bradford Mott. Conversational agents. In Munindar P Singh, editor, *The Practical Handbook of Internet Computing*. Chapman & Hall, 2004.

Fangtao Li, Xian Zhang, and Xiaoyan Zhu. Answer validation by information distance calculation. In *Coling 2008: Proceedings of the 2nd workshop on Information Retrieval for Question Answering*, pages 42–49, Manchester, UK, August 2008. Coling 2008 Organizing Committee.

Li Liu, Quan Qi, and Fangfang Li. Ontology-based interactive question and answering system. In *Internet Technology and Applications, 2010 International Conference on*, pages 1 –4, aug. 2010.

Hans P. Luhn. The automatic creation of literature abstracts. *IBM Journal of research and development*, 2(2):159 – 165, 1958.

Bernardo Magnini, Matteo Negri, Roberto Prevete, and Hristo Tanev. Is it the right answer? Exploiting web redundancy for answer validation. In *ACL*, pages 425–432, 2002.

Bernardo Magnini, Manuela Speranza, and Vikash Kumar. Towards interactive question answering: An ontology-based approach. In *Proceedings of 2009 IEEE International Conference on Semantic Computing*, pages 612 – 617, Los Alamitos, CA, USA, 2009. IEEE Computer Society.

J.S. Maritz. *Distribution-free statistical methods*. Science Paperbacks. Chapman and Hall, 1984.

Michael L. Mauldin. Chatterbots, Tinymuds, And The Turing Test: Entering The Loebner Prize Competition. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*. AAAI Press, 1994.

Mark T. Maybury. Question answering: An introduction. In *New Directions in Question Answering*, pages 3–18, 2004.

Diana G. Maynard. *Term recognition using combined knowledge sources*. PhD thesis, Manchester Metropolitan University, 2000.

Andrew McCallum and Ben Wellner. Conditional models of identity uncertainty with application to proper noun coreference. *Neural Information Processing Systems (NIPS)*, December 2004.

Xinfan Meng and Houfeng Wang. Mining user reviews: from specification to summarization. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, ACLShort '09, pages 177–180, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

Scott Miller, Heidi Fox, Lance Ramshaw, and Ralph Weischedel. A novel use of statistical parsing to extract information from text. In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*, pages 226–233, Seattle, Washington, 2000. Morgan Kaufmann Publishers Inc.

Tom M. Mitchell. *Machine learning*. McGraw-Hill Series in Computer Science. McGraw-Hill, 1997.

Ruslan Mitkov. *Anaphora resolution*. Longman, 2002.

Dan Moldovan, Sanda Harabagiu, Marius Pasca, Rada Mihalcea, Roxana Girju, Richard Goodrum, and Vasile Rus. The structure and performance of an open-domain question answering system. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL-2000)*, pages 563–570, 2000.

Raymond J. Mooney. Machine learning. In Ruslan Mitkov, editor, *The Oxford Handbook of Computational Linguistics*, chapter 14, pages 266 – 283. Oxford University Press, 2003.

Rafael Muñoz, Maximiliano Saiz-Noeda, and Andrés Montoyo. Semantic information in anaphora resolution. In *Proceedings of the Third International Conference on Advances in Natural Language Processing (PorTAL 2002)*, pages 63 – 70, 2002.

Vincent Ng. Unsupervised models for coreference resolution. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 640–649, Morristown, NJ, USA, 2008. Association for Computational Linguistics.

Vincent Ng. Supervised noun phrase coreference research: The first fifteen years. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1396–1411, Uppsala, Sweden, July 2010. Association for Computational Linguistics.

Vincent Ng and Claire Cardie. Identifying anaphoric and non-anaphoric

noun phrases to improve coreference resolution. In *Proceedings of the 19th international conference on Computational linguistics - Volume 1*, pages 1–7, Morristown, NJ, USA, 2002a. Association for Computational Linguistics.

Vincent Ng and Claire Cardie. Improving machine learning approaches to coreference resolution. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL2002)*, pages 104 – 111, Philadelphia, Pennsylvania, 7 – 12 July 2002b.

Dat P. T. Nguyen, Yutaka Matsuo, and Mitsuru Ishizuka. Subtree mining for relation extraction from Wikipedia. In Candace L. Sidner, Tanja Schultz, Matthew Stone, and ChengXiang Zhai, editors, *Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings*, pages 125–128. Rochester, New York, USA, The Association for Computational Linguistics, April 22-27 2007a.

Dat P.T. Nguyen, Yutaka Matsuo, and Mitsuru Ishizuka. Relation extraction from Wikipedia using subtree mining. In *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence*, pages 1414–1420, Vancouver, British Columbia, Canada, July 22-26 2007b. AAAI Press.

Dat P.T. Nguyen, Yutaka Matsuo, and Mitsuru Ishizuka. Exploiting syntactic and semantic information for relation extraction from Wikipedia. In *Proceedings of the IJCAI Workshop on Text-Mining and Link- Analysis, TextLink07*, 2007c.

Constantin Orăsan. PALinkA: a highly customizable tool for discourse annotation.

In *Proceedings of the 4th SIGdial Workshop on Discourse and Dialog*, pages 39 – 43, Sapporo, Japan, July, 5 -6 2003.

Marius Paşca. Organizing and searching the world wide web of facts – step two: harnessing the wisdom of the crowds. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 101–110, New York, NY, USA, 2007. ACM.

Marius Paşca. Outclassing Wikipedia in open-domain information extraction: weakly-supervised acquisition of attributes over conceptual hierarchies. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 639–647, Athens, Greece, 2009. Association for Computational Linguistics.

Isabella Peters and Paul Becker. *Folksonomies: Indexing and Retrieval in Web 2.0.* Knowledge & information : studies in information science. De Gruyter/Saur, 2009.

Massimo Poesio and Renata Vieira. A corpus-based investigation of definite description use. *Computational Linguistics*, 24(2):183 – 216, June 1998.

Massimo Poesio, Olga Uryupina, Renata Vieira, Mijail Alexandrov-Kabadjov, and Rodrigo Goulart. Discourse-new detectors for definite description resolution: A survey and a preliminary proposal. In Sanda Harabagiu and David Farwell, editors, *ACL 2004: Workshop on Reference Resolution and its Applications*,

pages 47–54, Barcelona, Spain, July 2004. Association for Computational Linguistics.

Simone Paolo Ponzetto and Michael Strube. Deriving a large scale taxonomy from Wikipedia. In *Proceedings of the 22nd Conference on the Advancement of Artificial Intelligence*, pages 1440–1445, Vancouver, B.C., Canada, 22-26 July 2007.

Ana-Maria Popescu and Oren Etzioni. Extracting product features and opinions from reviews. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 339–346, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

Yan Qu and Nancy Green. A constraint-based approach for cooperative information-seeking dialogue. In *Proceedings of the Second International Natural Language Generation Conference*, 2002.

Silvia Quarteroni and Suresh Manandhar. Designing an interactive open-domain question answering system. *Natural Language Engineering*, 15:73–95, 2009.

Karthik Raghunathan, Heeyoung Lee, Sudarshan Rangarajan, Nathanael Chambers, Mihai Surdeanu, Dan Jurafsky, and Christopher Manning. A multi-pass sieve for coreference resolution. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 492–501, Morristown, NJ, USA, 2010. Association for Computational Linguistics.

Altaf Rahman and Vincent Ng. Supervised models for coreference resolution.

In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 968–977, Morristown, NJ, USA, 2009. Association for Computational Linguistics.

Santosh Raju, Prasad Pingali, and Vasudeva Varma. An unsupervised approach to product attribute extraction. In *Proceedings of the 31th European Conference on IR Research on Advances in Information Retrieval*, ECIR '09, pages 796–800, Berlin, Heidelberg, 2009. Springer-Verlag.

Marta Recasens and Eduard Hovy. A deeper look into features for coreference resolution. In *DAARC '09: Proceedings of the 7th Discourse Anaphora and Anaphor Resolution Colloquium on Anaphora Processing and Applications*, pages 29–42, Berlin, Heidelberg, 2009. Springer-Verlag.

Marta Recasens and Eduard Hovy. BLANC: Implementing the Rand Index for coreference evaluation. *Natural Language Engineering*, 2011.

Marta Recasens, Eduard Hovy, and M. Antónia Martì. A typology of near-identity relations for coreference (NIDENT). In *Proceedings of LREC 2010*, pages 149–156, Valletta, Malta, 2010.

Verena Rieser and Oliver Lemon. Does this list contain what you were searching for? Learning adaptive dialogue strategies for interactive question answering. *Natural Language Engineering*, 15(1):55–72, January 2009.

Álvaro Rodrigo, Anselmo Peñas, and Felisa Verdejo. Overview of the answer validation exercise 2008. In *Proceedings of the 9th Cross-language evaluation*

forum conference on Evaluating systems for multilingual and multimodal information access, CLEF'08, pages 296–313, Berlin, Heidelberg, 2008. Springer-Verlag.

Bogdan Sacaleanu, Constantin Orăsan, Christian Spurk, Shiyan Ou, Oscar Ferrandez, Milen Kouylekov, and Matteo Negri. Entailment-based question answering for structured data. In *Coling 2008: Companion volume: Posters and Demonstrations*, pages 29 – 32, Manchester, UK, August 2008.

Boris Schooten and Rieks Akker. Vidiam: Corpus-based development of a dialogue manager for multimodal question answering. In Antal Bosch and Gosse Bouma, editors, *Interactive Multi-modal Question-Answering*, Theory and Applications of Natural Language Processing, pages 25–56. Springer Berlin Heidelberg, 2011.

Barry Schwartz. *The Paradox of Choice: Why More Is Less.* Ecco, 2004.

Wee Meng Soon, Hwee Tou Ng, and Daniel Chung Young Lim. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*, 27(2):521 – 544, 2001.

Martijn Spitters, Marco De Boni, Jakub Zavrel, and Remko Bonnema. Learning effective and engaging strategies for advice-giving human-machine dialogue. *Natural Language Engineering*, 15(3):355–378, July 2009.

Veselin Stoyanov, Claire Cardie, Nathan Gilbert, Ellen Riloff, David Buttler, and David Hysom. Coreference resolution with Reconcile. In *Proceedings of the ACL*

2010 Conference Short Papers, pages 156–161, Uppsala, Sweden, July 2010. Association for Computational Linguistics.

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. YAGO: A core of semantic knowledge. In *Proceedings of WWW-07*, pages 697–706, 2007.

Ying Sun, Paul B. Kantor, and Emile L. Morse. Using cross-evaluation to evaluate interactive QA systems. *Journal of the American Society for Information Science and Technology*, 62(9):1653–1665, 2011.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

Maite Taboada, Caroline Anthony, and Kimberly Voll. Methods for creating semantic orientation dictionaries. In *Proceedings of 5th International Conference on Language Resources and Evaluation (LREC)*, pages 427–432, Genoa, Italy, May 2006.

Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. Lexicon-based methods for sentiment analysis. *Computational Linguistics*, pages 1–41, 2011.

Yang Tang, Fan Bu, Zhicheng Zheng, and Xiaoyan Zhu. Towards Interactive QA: suggesting refinement for questions. In Nicholas J. Belkin, Charles L. A. Clarke, Ning Gao, Jaap Kamps, and Jussi Karlgren, editors, *Proceedings of the SIGIR 2011 Workshop on "entertain me": Supporting Complex Search Tasks*. ACM Press, 2011.

Pasi Tapanainen and Timo Järvinen. A non-projective dependency parser. In *Proceedings of the 5th Conference of Applied Natural Language Processing*, pages 64 – 71, Washington D.C., USA, March 31 - April 3 1997.

Masatsugu Tonoike, Takehito Utsuro, and Satoshi Sato. Answer validation by keyword association. In Vincenzo Pallotta and Amalia Todirascu, editors, *COLING 2004 3rd Workshop on Robust Methods in Analysis of Natural Language Data*, pages 95–103, Geneva, Switzerland, August 29th 2004. COLING.

David Traum and Staffan Larsson. The information state approach to dialogue management. In Jan van Kuppevelt and Ronnie Smith, editors, *Current and New Directions in Discourse and Dialogue*, pages 325–354. Springer, 2003.

Alan M. Turing. Computing machinery and intelligence. *Mind*, 59(236):433–460, 1950.

Olga Uryupina. *Knowledge Acquisition for Coreference Resolution.* PhD thesis, Saarland University, 2007.

Boris W. van Schooten, Rieks Op Den Akker, Sophie Rosset, Olivier Galibert, Aurèlien Max, and Gabriel Illouz. Follow-up question handling in the IMIX and Ritel systems: A comparative study. *Natural Language Engineering*, 15(1): 97–118, 2009.

Sebastian Varges, Fuliang Weng, and Heather Pon-Barry. Interactive question

answering and constraint relaxation in spoken dialogue systems. *Natural Language Engineering*, 15(1):9–30, 2007.

Suzan Verberne, Lou Boves, Nelleke Oostdijk, and Peter-Arno Coppen. What is not in the bag of words for *Why*-QA? *Computational Linguistics*, 36(2):229–245, 2010.

Suzan Verberne, Hans van Halteren, Daphne Theijssen, Stephan Raaijmakers, and Lou Boves. Learning to rank for *why*-question answering. *Information Retrieval*, 14(2):107–132, 2011.

Yannick Versley, Simone Paolo Ponzetto, Massimo Poesio, Vladimir Eidelman, Alan Jern, Jason Smith, Xiaofeng Yang, and Alessandro Moschitti. BART: A modular toolkit for coreference resolution. In *Companion Volume of the Proceedings of the 46th Annual Meeting of the Association for Compuatational Linguistics*, 2008.

Renata Vieira and Massimo Poesio. An empirically based system for processing definite descriptions. *Computational Linguistics*, 26(4):539 – 593, 2000.

Marc Vilain, John Burger, John Aberdeen, Dennis Connoly, and Lynette Hirschman. A model-theoretic coreference scoring scheme. In *Proceedings of the 6th Message Understanding Conference (MUC-6)*, pages 45 – 52, San Francisco, California, USA, 1995.

Ulli Waltinger, Alexa Breuing, and Ipke Wachsmuth. Connecting question

answering and conversational agents. *KI - Künstliche Intelligenz*, pages 1–10, 2012.

Dongsheng Wang. Answering contextual questions based on ontologies and question templates. *Frontiers of Computer Science in China*, 5:405–418, 2011.

Nick Webb and Bonnie Webber. Special issue on interactive question answering: Introduction. *Natural Language Engineering*, 15(1):1–8, January 2009.

Bonnie Webber and Nick Webb. *The handbook of computational linguistics and natural language processing*, chapter Question answering, pages 630 – 654. Wiley-Blackwell, 2010.

Daniel S. Weld, Fei Wu, Eytan Adar, Saleema Amershi, James Fogarty, Raphael Hoffmann, Kayur Patel, and Michael Skinner. Intelligence in Wikipedia. In *Proceedings of the 23rd AAAI Conference*, Chicago, USA, July 2008.

Liu Xiaoming and Liu Li. Chinese complex question analysis based on event extraction. In *International Conference on Computer Application and System Modeling (ICCASM)*, volume 8, pages V8–667–V8–670, 2010.

Weiqun Xu, Bo Xu, Taiyi Huang, and Hairong Xia. Bridging the gap between dialogue management and dialogue models. In *Proceedings of the 3rd SIGdial workshop on Discourse and dialogue - Volume 2*, SIGDIAL '02, pages 201–210, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

Zhongwu Zhai, Bing Liu, Hua Xu, and Peifa Jia. Clustering product features for opinion mining. In *Proceedings of the fourth ACM international conference on*

Web search and data mining, WSDM '11, pages 347–354, New York, NY, USA, 2011a. ACM.

Zhongwu Zhai, Bing Liu, Hua Xu, and Peifa Jia. Constrained lda for grouping product features in opinion mining. In *Proceedings of the 15th Pacific-Asia conference on Advances in knowledge discovery and data mining - Volume Part I*, PAKDD'11, pages 448–459, Berlin, Heidelberg, 2011b. Springer-Verlag.

Shubin Zhao and Ralph Grishman. Extracting relations with integrated information using kernel methods. In *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 419–426, Morristown, NJ, USA, 2005. Association for Computational Linguistics.

GuoDong Zhou, Jian Su, Jie Zhang, and Min Zhang. Exploring various knowledge in relation extraction. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 427–434, Morristown, NJ, USA, 2005. Association for Computational Linguistics.